JPRS-UCC-88-004
8 DECEMBER 1988

FOREIGN
BROADCAST
INFORMATION
SERVICE

JPRS Report

Science & Technology

USSR: Computers

JPRS-UCC-88-004
8 DECEMBER 1988

SCIENCE & TECHNOLOGY
USSR: COMPUTERS

CONTENTS

/9835

UDC 621.397.3

Contemporary Capabilities of Pattern Recognition in Applied Image Recognition

18630202 Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 87 pp 21-28

[Article by M. I. Shlezinger and G. L. Gimelfarb; first paragraph, authors' preface (presented as a footnote to the article's title)]

[Text] In the authors' opinion there is a rather obvious discrepancy between the theory of pattern recognition as it has been formulated to date and actual practice in the field of image recognition. It would of course be interesting to clarify the causes of this discrepancy and to point out ways of overcoming it. However, this article has a much less ambitious goal--to specify the highlights in the development of the discrepancy. This type of diagnosis will not lead directly to specific recommendations for eliminating the discrepancy. It is nevertheless a necessary step in solving the problem. The material included in this article is primarily in the form of a discussion, with its assertions being presented somewhat bluntly. These assertions should, of course, all be taken with a tacit qualifier on the order of "in our opinion," "perhaps," "as it appears to us," etc.

Introduction. In the past few years the practical demands imposed upon pattern recognition, particularly on image recognition, have increased significantly. This change marks the end of the transition from the prolonged "childhood" of this field of knowledge to its "maturity," which is just now beginning. Although pattern recognition has been geared toward one set of practical demands or another throughout the entire history of its development, the demands imposed upon the field have never been so specific or pressing as they are now. Previous ideas about the very broad sphere of the applicability of automatic recognition were formulated by the same circle of researchers who were studying the problem--often purely speculatively--while the actual practical demand for recognition systems has not been paid significant attention until very recently. The proposed user (generally referred to as "the consumer") perceived such systems as exotic albeit useful equipment that he could nevertheless do completely without. Today users need to perform tasks pertaining to the national economy in which recognition has become a real necessity rather than a cosmetic device. This has changed the demands imposed upon recognition substantially.

This change affected image recognition before it did other areas in pattern recognition. Because of this, image recognition quickly became separated from the more general problems of recognition. The first and for a long time the only example of the practical application of image recognition was the processing of photographs of the tracks of elementary particles in high-energy physics. More recently, the list of tasks related to recognition that have national economic importance has expanded significantly thanks to automation of the design, collection and analysis of aerospace video information during remote studies of the earlth's resources, ground and air navigation, visual inspection of industrial products, robotics, etc.

The demand for automatic pattern recognition has resulted in a sharp differentiation of this field of knowledge inasmuch as a careful examination of its specific applications reveals the inherent richness and uniqueness of each that cannot be noticed from a bird's eye view. As a result, if the fascination with the idea of a unified, common approach to recognition that characterized the early period in the development of the field has not been completely destroyed, at least there is a clear realization of the need to revise the fundamental formal concepts demarcating the bounds of the field that were formulated before such applications appeared. Differentiating the tasks of recognition will not, however, make it any easier to perform them. On the contrary, an exceptional complexity of most of these tasks, particularly the tasks of image recognition, has been discovered in this stage.

Modern recognition has already reached the stage where its further development depends on the degree to which the theoretical formulation and performace of tasks conform to the practical requirements of application regognition systems. It is also important to specify what pattern recognition can now offer in the way of satisfying urgent practical needs. We will therefore present a self-critical analysis of accepted fundamental concepts of the theory of pattern recognition and specify the formal framework in which all well-known recognition methods now lie.

We will study all the fundamental concepts as they relate to the application of image recognition and will try to specify which features of the application of image recognition have not yet been adequately formulated in the theory of pattern recognition.

Fundamental concepts of the theory of pattern recognition from the standpoint of applied image recognition. Pattern recognition as a scientific discipline was born in the course of the application task of inputting simple graphic images--written characters and geometric figures--into computers. The most distinctive feature of this task was the assigning of each image to a class based on the measurement of a large quantity of characters. Tasks from other fields of knowledge have the same feature. The attempt to formalize the common traits unifying all these tasks led to the creation of the modern theory of recognition. However, although those traits of the initial task that relate it to other tasks may have been its most noticeable traits, they were not of most essential ones. In view of this, the task of recognizing written characters did not remain in center stage--it moved to the periphery of the theory's application. The new applications of image recognition, though are close to the inputting of written characters into computers from the standpoint of their formulation, fall outside this area. We will examine this assertion in greater detail.

Three finite sets have become the main focus of modern recognition theory. These are the set T of the names of attributes (or, simply, the set of attributes), the set S of values of the attributes, and the set K of the names of classes of recognizable objects. The object is considered specified if for each trait $t \in T$ a value $v(t) \in S$ is indicated, i.e., the object is the function $v: T \to S$, whose domain of definition is the set of attributes T and whose range of values is the set S. Moreover, the function $k: S^T \to K$, which is termed the determinant rule, is specified in the set of all possible objects S^T. For each object $v \in S^T$ having the form $v: T \to S$, the determinant rule specifies the name $k(v) \in K$ of the class to which the object belongs. Thus the determinant rule k divides the set of objects into subsets (subdivision clases or classes) $V_\alpha \subset S^T$, which represent prototypes for solutions

$$\alpha \in K: V_\alpha = \{v: v \in S^T,\ k(v) = \alpha\}.$$

The set of classes is assumed to be finite and moreover to contain a small number of classes (from two to several hundred). A finite subset V, called a training set, is distinguished within the set of objects S^T. The training set together with the function $k_V: V \to K$ which is a narrowing of the function k into the set V, constitutes the training material $U = \{V, k_V\}$. Finally, the set K of functions of the type $S^T \to K$, which contains the function k, is also included in the examination.

The concepts that have been presented constitute the basis for formulating the most diverse tasks. Furthermore, in certain areas of recognition theory some of these concepts are not explicit as primary concepts but are instead specified on the basis of other concepts. In the statistical theory of recognition, for example, the set K of determinant rules is not specified. Introduced instead is a family of probability distributions on whose basis K is not postulated, but can be derived. Moreover, the concepts of training material and determinant rule acquire a definite statistical nuance. All of these peculiarities are not very essential in the context of the critical analysis that we are proposing.

For purposes of our discussion, we will examine one basic problem that has been formulated in terms of the concepts introduced in the preceding section. Given a known set K of functions and known training material $U = \{V, k_V\}$, find the function k^* for which its narrowing k^*_V into the training set V coincides with the assigned narrowing k_V of the function k in the same set.

In the specified problem, the function k is unknown, and the primal problem does not have a unique solution, i.e., there is a subset $K^* \subset K$ of functions whose narrowing into V coincides with the specified function. It depends on the specific form of the set K and the training material U. Whether there is such an interrelation of the sets K and U, in which the subset K^* will be "compressed" to the point where any function $k^* \in K^*$ is to a specified degree little different from any other function $K^{**} \in K^*$, is therefore a central question. The various recognition methods and algorithms differ with respect to the manner in which this yet nonformalized "compression" is specified. Moreover, the primal problem presented is specified by either the explicit or implicit introduction of additional constraints that make it possible to select the function k^* from the set K^* using a single method. These additional constraints are determined principally by the intuition and tastes of the investigator. When these constraints are not explicit but are instead expressed directly in the proposed recognition algorithm, the problem's solution is said to be heuristic. And if the additional constraints are specified in a strictly formal manner and unambiguously such that the recognition algorithm follows directly from these requirements, the problem's solution is said to be substantiated and strict. The set of such approaches and methods constitutes the theory of pattern recognition in its contemporary form.

We will further show that the specified formal framework of pattern recognition theory is distinguished by an excessive rigidity. Because of this rigidity, it is necessary when solving applications problems to distort their natural initial formulation so that they can assume a form that agrees with modern recognition theory. An alternative to this distortion is to solve each individual recognition problem apart from the existing theory as if one were starting from scratch in each specific case. Because this alternative will hardly arouse the enthusiasm of investigators, we need to explain exactly what the constraints of modern recognition theory are. First, the following three questions must be answered:

Is the concept of a recognition system as a device that classifies input data truly adequate?

Is the current method of specifying source data in the form of a list of attributes satisfactory?

Is the technology for obtaining the desired representation really equivalent to the training procedure that now occupies center stage of this technology?

We will examine all three of these questions.

Recognition as the classification of input data. A great deal of experience (both positive and negative) has been accrued in the solution of image recognition applications problems--optical readers have been designed and constructed, photographs of the tracks of elementary particles have been analyzed, drawings and sketches of electrical circuits and circuit boards have been recognized, multizone aerospace photographs of the earth's surface have been decoded, terrains have been reproduced in relief from stereoscopic pairs of photographs, etc. Without evaluating the end results of this research, it is nevertheless possible to state with certainty that not one of the application tasks has, in its initial formulation, been a clear cut image classification problem.

It is easy enough to think of tasks that could be formulated as classification tasks with potential practical value. Initial contacts with specific users show, however, that the final goal of recognition should not be the selection of the object's name from a small, previously specified list of names, but rather a more complex description of the object represented in the image. This description can be in the form of a number, set of numbers, list, graph of noted relationships, or simply another image.

Such information about an object should not be interpreted as a classification of objects. First (and foremost), there are many such classes and, more often than not, their number is not finite. Second, by considering the set of possible results of recognition as being identical with a set of class names without any structure, we soon lose the ability to use the a priori information that is contained in the applications formulation and that could be expressed in some metric in this set or in specified relationships between its elements. Because the set K of possible results of recognition is large, a rather large portion of the recommendations included in the general theory are not directly applicable.

Let a recognition algorithm be formulated that requires an instruction set containing at least one object from each class. Such a requirement is natural and perfectly easy to use given the condition that the output of the recognition system is information that can be expressed in elementary terms on the order of "no-yes." At the same time, as far as an applications task in which K is a potentially infinite set is concerned, such representative training sets are in principle impossible to achieve. Just as impossible to obtain directly are algorithms that in their intermediate steps calculate a set of numbers whose quantity is equal to the power of set K and that express the degree to which the recognized input object conforms to each of the possible output signals.

Thus, specific recommendations obtained without regard to the properties of the set K are quite far from being achievable in actual practice. In the case of images, this distance is so great that it can be described in terms of a rift between the theory of pattern recognition and the practice of image recognition.

To overcome this rift it is first of all necessary to carefully formalize the mathematical object that is supposed to be obtained at the output of the recognition system, i.e., to give the set K a structure that makes it possible to operate on its elements despite the fact that there may be an infinite quantity of such elements. Image recognition should be understood not as the classification of objects but rather as the determination of more complex, hidden characteristics that are not directly contained in the source data and are interpreted in terms of the component elements of a structural mathematical model of the object or set of objects presented in the image. The task of image recognition is thus fused with the more general task of understanding, i.e., interpreting the meaning, of an image.

In the structure of the set K the solutions being sought for $\alpha \in K$ may themselves become sets, for example, the set $A_\alpha = \{E_\alpha, R_\alpha\}$ of names $e \in E_\alpha$ of individual objects of the set represented in the image and relationships $r_{\alpha.\lambda}(\cdot) \in R_\alpha$ between these objects:

$$R_\alpha = \{r_{\alpha,1}(e_1), r_{\alpha,2}(e_1, e_2), \ldots, r_{\alpha,\Lambda}(e_1, \ldots, e_\Lambda) : e_1, \ldots, e_\Lambda \in R_\alpha\},$$

where λ is the maximal arity of the relationships. In the simplest case ($\Lambda = 1$), we have the task of classifying the components of a composite object (the individual parts are not connected). This task is rarely encountered in pattern recognition. In all other cases ($\Lambda \geq 2$), the complexity of performint the recognition task increases sharply as the complexity of the relationships R_α increases.

The nature of the set of attributes. Usually in pattern recognition the properties of the set T of attributes are not defined concretely, i.e., T is considered a finite set of indexes (names of attributes). Such a formalization of the set of source data essentially unifies the data known at the moment when the applications task is formulated. Thus, in image recognition the set T is not an abstract set of names of attributes, but rather a set of digital signals that has a distinctly defined mathematical structure that makes it possible to speak of the existence of specified operations and relationships within this set. This structure is especially pronounced in the raster method of representing images. The most important visual concept, such as symmetry, connectivity, dimension, orientation, are simply meaningless outside of this structure. For this reason, research done without taking the structure into account will not be specific enough. Also, ignoring the real structure of the set of attributes that is adequate for image recognition results in the secret involvement of other structures that seam logical but that do not have any relation to image recognition.

Thus, in its time, the set came to be identified with the set of coordinates of a multidimensional Euclidean linear space of signals; and thus the set of recognizable objects came to be identified with the set of points in this space. The sharp remarks of M. Minsky and S. Peypert concerning this subject are well known. They maintain that nothing has done more harm to the field of image recognition than the involvement of multidimensional geometric analogies. The point is that simple subsets of points in a multidimensional space (a sphere, the intersection of half-spaces) do not correspond at all to visual concepts that are elementary in content and intuitively very simple (the line, polygon, and circle). On the other hand, operations that are simple to assign in a multidimensional space do not necessarily have simple visual interpretations. Therefore the involvement of multidimensional geometric analogies in image recognition has not yielded the expected results. In fact, they have ad a disorienting effect. When the set of attributes is considered identical to the coordinate axes of a multidimensional space, the initial structure of this set is destroyed, and the properties of images that are completely obvious in the initial applications formulation of the task become hidden as the task is formalized.

Although the necessity of examining the structure of input and output data has not yet been adequately addressed in recognition theory, these structures are already being used in practical applications. This means that practical algorithms and programs are in fact already in the form of converters having a set of objects with a family of relationships specified in it at their input and output. The set of existing algorithms of this type has been termed the structural approach to pattern recognition, even though this approach has yet to form a theory. The reason for this is that a system of fundamental formal concepts that would permit the unique formulation and solution of a task has yet to be developed. For this reason, only uncoordinated methods connected by some intuitively understood but unformalized generality exist here. The absence of any theoretical generalization and comprehension of this type of algorithm means that although a structural approach to recognition does exist in a way, the accomplishment of one task does not make it the least bit easier to accomplish other tasks. The mass demand for recognition systems that is now being felt in practically all sectors of the national economy cannot be satisfied as long as this rift between theory and practice exists.

The role of training in image recognition. As before, we consider a recognition system to be the implementation of the function $k:V \to K$ (mapping the set V of recognizable objects onto the set K of results of recognition). In an earlier section of this article we have already concluded that both these sets should be formalized more carefully than has been done up to now. We will now examine a technique for finding the function k and will question the established idea that so-called training in recognition is the most important element of this technique. The training consists of finding the function k, which belongs to the previously specified set K of functions and which assumes the prescribed values in the specified training set of objects.

At the present time most of the theoretical research in pattern recognition is concentrated around the topic of training. The degree to which it has been developed is such that it is time to state that the topic of recognition is not exhausted by the training problem alone. At least two fundamental difficult problems exist, each requiring a solution. The first problem is that when applications tasks are being performed the information that should be specified as given during the solution of the training problem is, as a rule, unknown. The second problem is that of constructing an algorithm to compute the already known function k, which is found in the training process.

The set K of functions is the most important element of the source data for the training task. As is well known, this set should be sufficiently "compressed" that the small amount of additional information contained in the training material is sufficient for a unique specification of the required mapping of the set V of objects onto the set K of results of the recognition. The problems entailed in finding the set K are outside the theory of training and are placed completely on the researcher performing the recognition applications task. To implement his idea about a suitable (sufficiently narrow!) set K in a particular case, a researcher must study his applications task in enough depth that it will as a rule not be difficult for him to also directly determine the required function itself. In other words, he must simply make the training task unnecessary.

Introducing the training procedure into the technique of obtaining the function k means only that the researcher does not have to fully complete this process of researching the application and that sometimes the final stages of the study become so obvious that the task can be formally performed by the recognition system itself. Hidden behind this seemingly pleasant situation, however, is the fact that the applications task should be studied almost to the end. This much less attractive bit of information is discovered in the very beginning stages of one's acquaintance with the applications task. Not studying the applications task deeply enough leads to the specification of too rich a set K of functions claiming correct recognition. This in turn leads to an impossibly lengthy training period. Even if this is not so, no amount of training will free the designer of a recognition system from the necessity of finding a specific algorithm for implementing the function k after it has already been uniquely specified, whether in the training process or by some other method.

At present, thanks to the advanced theory of computations, the fundamental difference between the unique specification of a function and the algorithm for computing it is well recognized. Inasmuch as recognition theory has generally used model, artificial examples rather than real tasks as a test polygon of its recommendations, sets of functions that have not been fundamentally difficult to implement have generally been selected as the sets K.

For a long time it was assumed that there exists some very simple and easily implemented recognition function, for example, a function that was linear with respect to its input variables; and the crux of the recognition problem was seen to be that this function was unknown. A myth about omnipotent training that would settle the problem of recognition thus evolved. In reality, the principal difficulty is that the required function, unknown though it is, does not belong to any class of well-studied and therefore simple mathematical functions. Therefore designing an algorithm to implement such functions is fraught with fundamental computational difficulties. The training procedure does not in fact eliminate these difficulties at all: the construction of the training recognition system not only does not free the designer from developing an algorithm to compute k, it also requires him to develop a unique algorithm to compute any function from the set K, which is a much more complex task.

Unfortunately, it is only possible to understand how to construct the set K and algorithms executing the functions of this set by carefully studying each specific applications task and performing the largest possible number of such tasks. Thus the strict practical requirements of pattern recognition are not o only a stimulus but also a necessary condition for its development. Even if these external requirements did not exist, they would have to be thought up in order to develop pattern recognition in its purely scientific aspect!

Applications of image recognition. Within the field of image recognition one can identify at least three approaches to applcations that differ from one another from the standpoint of task formulation and performance. These are the recognition of images created by man for interaction with a computer; the recognition of images of artificial manufactured objects created (for example, automated visual product inspection); and the recognition of images of natural objects, such as natural formations.

Image recognition as a method of man-machine communication. Until recently the standard task in this category was the creation of reading automatons for inputting texts into computers. In the last few years the applications sphere, which very much needs this type of recognition system, has expanded significantly on account of the need to recognize the most diverse types of information in sketch and graphic form: circuit board sketches, basic diagrams of computer equipment, cartographic information in the broad sense of the term (not only graphic maps but any schematic representation of the location on a plane of interconnected objects), etc. In all of these cases, recognition entails obtaining a description of an object that exists either in reality or in the human imagination based on a conventional representation of the object in the form of a sketch created by man. At first, attempts were made to accomplish the task of recognition without introducing any type of regulatory constraints on exactly how human conceptions should be represented in a drawing.

Particularly intensive searches for this type of algorithm were conducted for the purpose of recognizing hand-written characters. There were numerous announcements of the successful accomplishment of these tasks, but the reported successes turned out to be in words alone. The main conclusion derived from the research conducted is that of the practical value and viability of only those recognition systems that are based on absolutely unambiguous and formal regulation of the form in which the information to be input into a computer should be put down on paper by man. This conclusion is addressed not only to developers of recognition systems but, to a lesser degree, to system users as well. It can be expected that in both cases this conclusion will not be accepted painlessly.

The transition from remote and thus romantic, mist-shrouded goals of creating devices that understand people to the closer, prosaic goals of simplifying and accelerating the process of inputting graphic materials into computers requires changing the very nature of research in this area. This is something that, of course, not all researchers can do. As for the users of recognition systems, disappointment awaits them. If a user wants his sketch or other graphic document to be correctly understood by a computer, the document must be prepared in the proper manner. This disappointment will be all the stronger because specialists in pattern recognition have for years been promising the user something quite different.

The realization of the need to regulate images undergoing automated recognition directly changes the nature of the tasks that must be performed in designing these recognition systems. For the sake of illustration we will use the analogy of designing translators from an algorithm language.

First, the much-loved criterion of minimum probability of erroneous recognition is disappearing from use in pattern recognition. Just as a translator should process any syntactically correct sentence correctly, so should an image recognition system provide a correct interpretation of any image satisfying all regulatory requirements. Of course, there are cases where translators have been produced that make errors even in syntactically correct source texts. However, the requirements imposed on translators are such that any error discovered is grounds enough to deem that translator unfit for service and send it back for modification.

If the need to regulate input images for a recognition system has been recognized, then it should also be required that the probability of the incorrect recognition of a correct image be reduced not merely to a low number but to zero, i.e., the possibility of such error should be eliminated. The most important features of a recognition system is thus not its error probability (which equals zero) but rather the convenience, naturalness, and ease with which it meets those requirements that a sketch must satisfy so as to guarantee error-free recognition. Thus, image recognition arrives at the same criterion as translators, which the main feature is convenience of the input language rather than error probability, which should be zero.

Moreover, just as a translator is a program with the set of all possible sentences as its domain of definition rather than only syntactically correct sentences, so too is a recognition system a program that should be based on the set of all possible images rather than only correct ones. Diagnostics should be output for incorrect images, i.e., it should be precisely indicated where in the sketch the violation occurs. True recognition should be executed for correct images, i.e., a description of the object or situation depicted in the sketch should be formulated.

And finally, although the translator is the most important component when the user inputs ideas about a required program into a computer in algorithm language, it is nevertheless only one of the links in a rather rich production line. The same is true of an image recognition system. It is only one link in a production line for inputting the information contained in an image; this production line must also, among other things, provide for making online revisions in input data that are syntactically incorrect, checking the final results, and when necessary revising these results.

The process of designing an image recognition system while performing an applications task thus turns into the process of designing a production line to input images into a computer (preparing images, recognizing them, and checking and editing source, intermediate, and resultant data). All of these links must be intereconnected so as to minimize not the probability of erroneous input but rather the cost of error-free input.

Such has been the evolution of views on the problem of image recognition in one particular but rather widespread case. As a result of this evolution, which is far from having been completed, the romance and mystique of recognition systems is gradually disappearing and is being replaced by a rather prosaic certainty in the efficiency of such systems.

We will note that the more than 25 years of experience that have been accured in the design of reading automatons has shown that the only projects to "survive" have been those that from the very outset were geared toward creating an entire production line for inputting text documents written in characters into a computer and that began from a rather strict regulation of the forms of documents, methods for filling them with text, allowable type, shapes of characters, etc.

As experience in the development of a number of existing systems for recognizing sketch and graphic images has shown, the requirements involved are fully achievable in practice.

Some other tasks of image recognition. Although the possibilities of regulating input data in the case of natural and artificial, manufactured objects are less obvious, the basic principles described earlier also apply here to a certain degree. It is precisely the regulation of data, whether explicitly or implicitly, that makes it possible to construct mathematical models characterizing the structure of the relationship betweeen the signals measured and objects (the desired results) of recognition and to obtain image recognition algorithms based on these models.

By way of an example one can mention the series-produced recognition systems for visual inspection of the shape of tablets on a pharmaceutical conveyer line or research systems for analyzing the images of three-dimensional scenes consisting of polyhedra with uniformly colored surfaces. Narrowly specialized recognition algorithms designed for a strictly limited set of classes of recognizable objects are most often executed in such systems. The dimensionality of the objects is important--planar objects are more easily recognized than are three-dimensional ones.

In the case of planar objects that do not overlap and are within the field of vision of the image receiver, each individual image characterizes the form and radiometric properties of the surface of each object with sufficient completeness. Any point on the object's surface is identical to the analogous point on the image, which makes it possible (as in the case of graphic materials) to speak in terms of the practical identity of objects and images. The situation is different in the case of three-dimensional objects. Here each individual image contains only that portion of the visible surface that falls within the receiver's field of vision. For this reason, images of all the objects from many angles of vision are necessary to obtain a sufficiently complete reconstruction of the entire set of objects observed and their relationships in space.

The relative successes that have been achieved in the automated recognition of individual images in scenes composed of uniformly colored polyhedra are explained by the specific features of these bodies, which do not have parallels among bodies with curved surfaces: abstracting from the object's shadows, there is a unique simple interrelationship between the geometric primitives of a polyhedron (its faces, edges, and vertices) and the local characteristics of its image signals (polygonal sections with approximately constant signal values and rectilinear boundaries between adjacent sections with different signal values along both sides from the boundary and the boundaries' intersecting points). Unfortunately, this type of interrelationship, which simplifies the construction of mathematical models of objects and signals, is the rare exception in image recognition tasks.

Tasks of recognizing the images of natural three-dimensional objects are the most difficult to formalize. It is thus not surprising that the performance of such tasks has still not progressed beyond the scope of initial exploratory research. This is primarily due to the absence of effective mathematical means of formally representing knowledge about recognizable objects; and, in a number of cases, it is also due to the definite "fuzziness" (disorderliness) of the knowledge.

It is well known that certain tasks related to image interpretation are accomplished in different ways by different observers, even if they are specialists with the same background. Simply compare maps of linear elements compiled by different geologist-interpreters during the geologic decoding of the exact same photographs of the earth's surface. The main efforts in this area have therefore been directed toward working out a formal apparatus for representing knowledge. As has been mentioned earlier, the concept of training is qualitatively different from those simplified concepts that have evolved in modern pattern recognition. It consists of the selection and refinement of a specific structure for representing knowledge about the training material. If the knowledge is represented in the language of graphs of observed relationships or generative grammars or, finally, block diagrams like the frames used by M. Minsky, then such training will consist of (1) a search for the best (in one sense or another) subset of nodes and arcs of the graph that corresponds to the specific class of objects (2) the selection of an alphabet of primitives and their formulation rules for grammar, or (3) the introduction of internal structural connections in the frame.

The resultant optimization takss are so complex that it is necessary in each individual case to look for approximate heuristic solution techniques that are suitable for that case alone. The direct construction of the desired representation during the task formalization process is a substantially more achievable alternative. It is precisely in this manner that most image recognition tasks are accomplished.

Diverse mathematical models of images, recognition objects and recognition algorithms based on such models have already been constructed. These results are outside the set of topics being examined in this article. For our purposes, the only thing of importance is that these results are almost never connected with the theory of pattern recognition and do not form a proper theory of image recognition in the sense that they have no common axiomatic basis and do not provide uniform methods of formulating and accomplishing different applications tasks.

The task of decoding multizone aerospace photographs of the earth's surface may be specially singled out among the tasks related to recognizing the images of natural objects. The practical importance of the results of this task for many sectors of the economy and the great volumes of incoming data have caused researchers and consumers to focus a great deal of attention on this field. The current level of the theory and practice of recognition is still far from formalizing the formulation and performance of tasks that are comparable in complexity to the semantic interpretation of such photographs. For this reason, virtually all existing exploratory work in this direction has been done for those natural objects that can be fully defined by a small number of attributes that clearly set these objects off from the background of other objects (let us say the color, or in the more general case, the intensity of the radiation in several bands of the spectrum for all of an object's points). Because the interpretation of photographs requires extensive efforts on the part of humans and because the results obtained by different groups of observers are far from constant, interactive (man-machine) hardware and software interpretation complexes in which computer processing is directed primarily toward accelerating, easing, and expanding the potential of human interpretations are now being created.

The increase in consumer interest in automated image processing systems, which has occurred in recent years, is due to the fact that much of the recognition of the images of real objects today and in the near future can only be accomplished through man-machine data analysis. Automated image processing systems primarily implement conversions of the "input image-output image" type, which expand the possibiltities for visual interpretation of photographs by human beings--change in the color scale, emphasis on the contrast of individual details, isolation in the plane of the photograph of sections that are relatively uniform in the spectral composition of their signals, etc. Individual tasks related to isolating uniform sections are now formulated and performed as traditional recognition tasks and recognition training, where signals belong to one class satisfying accepted uniformity criteria and the structure of the interrelationships of signals and classes in the plane of the photograph are not taken into account. However, such tasks are only of interest in the very early stages of man-machine photograph analysis; all other formulations of the tasks of image processing and recognition are already far removed from such "structureless" classification of signals. Unfortunately, it is still premature to speak of any common theoretical technqiues for achieving uniformity in the formulation and performance of the tasks related to recognition of images of natural and artificial objects.

Conclusion. The strict practical requirements that have been imposed upon pattern recognition have made it necessary to reassess the relationship between modern pattern recognition theory and the practice of image recognition. The relationship seemed completely satisfactory before these requirements appeared. It is now clear that the theory of pattern recognition, as it has been formulated up to now, ignores certain very important aspects of image recognition. The rift that has developed should be overcome by reexamination of techniques that are already being used. The creation of a fully adequate theory of image recognition is necessary in view of the fact that the mass demand for image recognition systems can no longer be satisfied by individual production of recognition algorithms.

It is however necessary to caution the consumer about the illustions that evolved in the early period of recognition. Hopes for a wonder-working recognition system should be replaced by a sober understanding of the true capabilities of such systems, just as happened earlier with regard to the use of computer technology in general.

12794/9835

Selected Software: Concise Descriptions

18630204a Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 87 pp 83, 115

[Article: "Software Available in Ukrainian SSR Academy of Sciences Software Resources"]

[Text] Sirodzha, I. B., Diskant, V. A., Fursa, V. I., Yeremina, L. A., Krylov, Ye. M. and Minevskaya, T. V. "KOD-3 Applications Package for Recognition and Classification of Objects Having Various Types of Features", (YeS1033, FORTRAN-IV, PL/1, YeS [Unified Series] Operating system, version 4.1), Aviation Institute, Kharkov, 1985, 456 pages, inventory No P-6302.

The KOD-3 applications package is designed for the automation of decision making by means of the prompt construction of structural analysis rules for the classification of objects according to various types of data in the following tasks: the recognition of objects in teaching and self-teaching, and the analysis of expert evaluations and group sampling data. A new structural analysis method of classification has been implemented in the KOD-3 package that can automate the process of classification data processing and optimize the decision-making rules obtained.

KOD-3 can be used by scientific research organizations, industrial enterprises and computer centers for the prompt construction of decision-making rules in automated process control system and automated design system tasks, the analysis of expert evaluations, and technical medical diagnosis.

The KOD-3 package contains 172 programs organized into an overlay structure and contains 15,400 commands. It requires 290K of main memory. The program functions in the batch processing mode.

Fedunets, P. D., Perevoznyuk, L. S. and Fedunets, A. D. "Software System for Simulation of Operation of Transport-and-Warehousing Systems", (YeS1022, FORTRAN-IV, YeS Operating System, version 4.1), Technological Institute of the Food Industry, Odessa, 1985, 105 pages, inventory No P-6300.

This software system is designed for computer simulation of the operation of transport-and-warehousing systems and is used in their studying and design.

The software system's area of application is limited by the transport-and-warehouse system's type of structure and the number of its elements (up to 1000). The software system consists of a main program and 19 subroutines; it implements the functional algorithm for standard elements and forms the structure and assesses the quality of the system's operation. The simulation method was used in writing the modeling algorithm. This software package is distinguished by its ability to optimize the structure and operating modes of the system's lements.

The software system has 1237 commands. It operates through batch processing and requires 250K of main memory.

Akhlamov, a. G. and Kondratenko, A. A. "ADOKM--Automatic Documentation for Models", (YeS computers; ASVT [Modular Computer Technology System], M4030 (M4030-1), PL/1, YeS Operating System, version 6.1), Institute of the National Economy, Odessa, 1983, 68 pages, inventory No P-6296.

ADOKM is designed for the computerized construction of graphics flowcharts representing the structure of models using simulation language GPSS. The program can be used for simulations in GPSS. There are restrictions on the amount of initial data, the program in GPSS must be syntactically correct, and a subset of the GPSS language is used.

The program consists of two internal modules--NDOK1 and NDOK10--which perform the following functions: input of the source text in GPSS and the construction of a graph of the model; and the construction, disposition and printing of the flowchart's graphics. Standard methods and algorithms are used for the syntactical analysis of texts and the organization and retrieval of data that have the structure of an oriented binary graph. There are no analogs of the ADOKM program for simulation languages.

The program has 840 commands. It uses batch processing and requires 100K of main memory.

Babichenko, G. D., Pershin, A. V. Shishlov, V. I. and Yurkov, V. M. "DISVEDOR--Interactive Document-Oriented Development System," (YeS1022, PL/1, assembly language, YeS Operating System, version 4.1), Gorsistemotekhnika NPO [Scientific Production Association], Kiev, 1983, 236 pages, inventory No P-6284.

DISVEDOR is designed for the creation and operation of interactive applications systems oriented toward the automated development and management of text documentation. The following are the system's main functions: generation, loading, and correction of a database; editing of information; output for printing in standard form; authorized connection of users and access to information; and the management of development according to a specified scenario. The implementation of the last function, as well as the system's orientation toward the non-professional computer user, are distinctive features of the system. The software consists of interactive and auxiliary systems.

The software system contains 200 commands. Hardware restrictions: YeS7066 video terminal, single-terminal mode. Requires 130K of main memory. Users of various categories are offered the following: a simple mnemonic instruction language, an extended instruction and message system, facilities for forming a scenario--a description of the user's method of working and teaching facilities. The system is self-documented.

8831/9835

UDC 510.6:681.3.06

Making Information Systems Intelligent

18630204b Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 87 (manuscript received 8 Jul 87) pp 84-92

[Article under the rubric "Databases and Information Retrieval Systems," by F. I. Andon, E. G. Zakharova, V. A. Reznichenko and A. Ye. Yashunin]

[Text] Introduction

The area of application of computers is extremely broad now, and the appearance of inexpensive personal computers is making the presence of computers on school desks, on the desks of administrators, alongside the cabinetmaker's bench and at home an ordinary thing. Nevertheless, current computers have one shortcoming: There is between them and people wanting to use them independently for solving their own problems a wall of incomprehension, which can be surmounted, as a rule, only by means of programmers or training in programming. The current language barrier between computers and the end user must be eliminated. The solution of this problem requires the development of new untraditional systems facilities capable of taking upon themselves to an ever greater extent the execution of tasks previously solved exclusively by a human being. Research on artificial intelligence formed on the basis for the development of these facilities, and it has become obvious now that the successful solution of the problem is impossible without enlisting the ideas and methods developed in this field. [1-3]

The final goal of this research is the development of a new information technology utilizing the characteristic methods of artificial intelligence and, above all, the ability to operate with knowledge approximately the same as a human being does in the performance of creative work. The knowledge base, in which a human being's knowledge about his subject area is represented in computerized form, is an integral component of the systems under development using this new information technology. It is precisely the presence of this knowledge that makes it possible for the system to solve successfully new problems that previously were the prerogative of man.

This paper is devoted to a logical analysis of the intelligence capabilities and problems in the development of one class of systems under development within the framework of the new information technology: intelligent automated information systems.

By an automated information system (AIS) we will mean an organization-software-hardware system whose main purpose is the gathering, reliability-checking, storage, modification and output of information. The information base (IB), containing in one form or another the information users need, is the central component of an AIS. The neutral term "information base" is used--in order to emphasize the main purpose of this component of an AIS--the representation of information in a computer. At various times and in various contexts this component has borne the name data files, data libraries and databases, and recently the term "knowledge base" is being used ever more frequently.

In this paper the authors regard an AIS first of all as a tool for the registration of information--knowledge concerning a subject area--and as a means of obtaining and verifying new knowledge, achieving a unique understanding of the essence of the problems present in the given subject area, and forming the necessary basis for the precise formulation and solution of problems. Principal attention is paid to a logical analysis of the capabilities of an AIS with regard to the representation of knowledge and its use and to logical problems in the creation and development of an AIS. Questions such as the style and means of communicating with the user, architecture and efficient implementation, organization, hardware and a number of other aspects of no less importance for the development of a viable, practical AIS have been deliberately left outside the scope of this article.

A General Look at Problems of Representation of Knowledge in AIS

The developer of an AIS, for the sake of his own practical interests, isolates from the overall picture of surrounding reality a certain fragment constituting his subject area. In the course of studying the subject area a certain mental image of the subject area forms in the consciousness of the developer of the AIS, and as this mental image becomes more clearly defined, it is transformed into knowledge--a unified and systematized set of judgments [4] representing in the opinion of the developer of the AIS, the real state of affairs. We will call the representation of knowledge about a subject area by a formal language in an AIS a knowledge base. The actual process, of studying a subject area and forming a mental image of it, concluding in the construction of a knowledge base, is called conceptual modeling.

Thus, three separate, though interrelated, entities take part in the process of conceptual modeling: the subject area, its mental image, and the knowledge base. In the ideal case they should fully complement one another. However, in practice this is far from always the case. No methods exist that would formally and automatically make it possible to verify or guarantee such adequacy.

From the authors' viewpoint there are two directions for improving the efficiency, reliability and adequacy of conceptual modeling:

1. The selection and development of a precise conceptual system for the conceptualmodeling of a subject area. The concepts selected must have sufficient generality and breadth of application, reflect essential aspects and features of objective reality and have a simple, clear meaning.

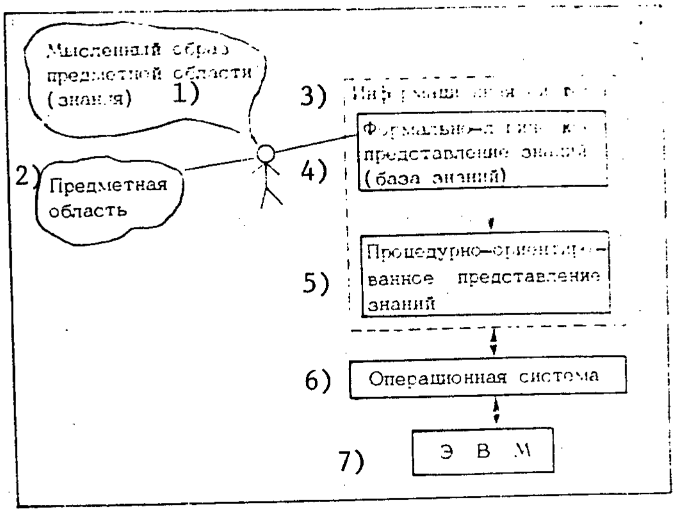


Figure 1.

Key:

1. Mental image of subject area (knowledge)
2. Subject area
3. Information system
4. Formal-logic representation of knowledge (knowledge base)
5. Procedure-oriented representation of knowledge
6. Operating system
7. Computer

Development of Coneeptual System

It is natural to begin with a precise definition of the concept of a subject area. An ISO report [5] gives the following formulation of this concept.

The subject area is those essences of interest that have existed, do exist or could exist at some time. Here an essence can be anything at all that is of interest to the developer of an AIS and is perceived by him as something existing (having existed or able to exist) regardless of whether it exists physically, in abstract form or only in the imagination.

We will call the state of the subject area the set of all essences that for whatever reason are perceived by the developer of an AIS as related. The concept of the state is always singled out when the possibility of changes in the subject area is permitted. The appearance and disappearance of types of interrelationships, etc., are examples of such changes. Thus, a subject area's existence can be thought of as the process of successive transformations from one state to another.

Conceptual Schema and Conceptual Base

The developer's knowledge of a subject area means his knowledge of those essences that have existed (if this is necessary), do exist or could (should) exist. He knows the subject area if he knows its possible states, the possible transformations from state to state and the subject area's current state. It is precisely these three kinds of knowledge that he must represent in the form of a knowledge base.

This "division" of knowledge into three kinds yields the natural division of a knowledge base into three parts. The first two kinds of knowledge are of the greatest (essentially, the only) interest at the stage of conceptual modeling. By possessing them the developer of an AIS will essentially understand what the subject area represents in general, what can and cannot occur in it, and what general laws, rules, restrictions, etc., are at work in the subject area and control its development. It is precisely these two kinds of knowledge that he will strive to obtain first of all and to register in the knowledge base.

The conceptual schema is the formal representation in the knowledge base of the set of judgments characterizing the set of possible states of the subject area and the set of possible transformations of the subject area from one state to another.

The conceptual base (KB) is the formal representation in the knowledge base of the set of judgments characterizing the concrete current state of the subject area.

With regard to the nature of the judgments that can be represented in a conceptual schema and conceptual base, the former are often called rules, or, when it is necessary to distinguish judgments themselves from their formal representation, rules and axiom rules. The latter, valid perhaps for just one concrete state, are called facts or, when making the same distinction as in rules, facts and axiom facts.

Thus, any subject area can be perceived as a set of M possible states and a set of R possible transformations from one state to another. Accordingly, a conceptual schema characterizing both these sets can be divided into two parts describing M and R. The part describing M is called the description of the subject area's statics by means of static rules and static axioms. These exioms and rules postulate what can or cannot be in any of the subject area's states regardless of the history of its development and regardless of how and from what state the subject area passes into the state in question (which explains the name).

The part of the conceptual schema describing set R is called the description of the subject area's dynamics by means of dynamic rules, transformation rules and dynamics axioms. These axioms and rules postulate possible changes (transformations) of subject area states and into what kind of state a subject area can change from a given state or after a sequence of given states, and characterize the possible development of the subject area.

Dynamics of Subject Areas

The next step along the road of defining more precisely intuitive concepts concerning a subject area should be a clarification of the question of the nature of the elements of sets M and R.

The simplest case will occur when the possibility of a transformation is determined totally by the initial, m_1, and final, m_2, states (m_1, m_2 M) of any transformation regardless of the previous history of state m_1. In this case each transformation can be represented by a pair of states, $\langle m_1, m_2 \rangle$, and set R will be a binary relationship in M:

$$R \subseteq M \otimes M,$$

so that $\langle m_1, m_2 \rangle \in R$ will mean "a transformation into state $m_2 \in M$ is possible from state $m_1 \in M$" and $\langle m_1, m_2 \rangle \notin R$ will mean "a transformation from $m_1 \in M$ into $m_2 \in M$ is not possible." In the more complex case, the possibility of a transformation from a certain state $m_k \in M$ into $m_{k+1} \in M$ can depend on the sequence (arbitrary length) of states, $m_1, \ldots, m_{k-1} \in M$, preceding state m_k.

Statics of Subject Areas

Our next task is to attempt to define more precisely the state of the subject area, i.e., to attempt to define more precisely the nature of the elements of set M. Since the state of the subject area is defined as a set of essences, this problem is inseparable from defining more precisely the concept of essence. It is impossible to determine beforehand what the developer of an AIS will consider an essence. Nevertheless, it is probably possible to isolate the most general concept of an essence relevant to any subject area. This is the mathematical concept of a class or set, which makes it possible to add to our lexicon an extensive class of set theory concepts and structures for their formal representation with the precise axiomatization of the most fundamental features of these concepts. At the same time it becomes possible to transform the vague concept of essence into a precise mathematical entity and to use a strict formal device--axiomatic set theory [6-8]--for representing it and studying its features and interrelationships.

We will assume that at some stage the developer of an AIS arrives at a mental representation of a subject area in the form of a set theory structure that can be strongly correlated (identified) by him with reality, and that his next task will be to construct an axiomatic formal theory within the framework of some kind of logic, and that this structure (equivalent in his understanding to the subject area) will serve as a model for this logic.

In this structure the most elementary essences of the subject area whose internal structure the developer of the AIS is not interested in will be (mentally) represented as individuals [9]. The presence of general features in them will be expressed in the definition of various sets of individuals and the relationships between essences will be represented by relations which in turn will be mental images of other more complex essences. Correspondingly, these complex essences can have their own features and enter into relationships with other (simple or complex) essences, and this will be expressed in the definition of a set of sets, relations of relations, etc. The representation of a subject area as a set theory structure--$\langle S, T_1, \ldots, T_n \rangle$ (where S is a set of individuals and $T_1, \ldots, T_n$ represents relations), whose components form the content of concepts [4] relevant to this subject area is the first step toward a formal description of it. A judgment concerning the membership of one component of the structure--S or T_i--in a certain essence (simple or complex) can naturally be called an (elementary) fact. And a set of facts perceived together as true can be called a mental image of the state of the subject area.

Thus, set M of states of a subject area can be represented as a set of structures, $\{\langle S, T_1, \ldots, T_n\rangle_i\}$, where each state, $\langle S, T_1, \ldots, T_n\rangle_i$, represents the content of concepts relevant to the subject area. Stable relations between concepts will be expressed in corresponding relations between the contents of concepts occurring for each $m = \langle S, T_1, \ldots, T_n\rangle_i \in M$. The representation of judgments concerning these relations in a conceptual schema will give us statics axioms for which M will appear as a set of models. R will be the set of models for dynamics axiomx, and in the simplest case $R \leq M \otimes M$. Finally, the set of propositions concerning the membership of essences in sets S, T_1, ..., T_n of a certain specific state $\langle S, T_1, \ldots, T_n\rangle_i \in M$, will represent a description of the specific current state of the subject area--the conceptual base.

This treatment of the general concepts associated with a subject area, its mental image and the representation of this image in the form of a knowledge base makes it possible to cover a very wide range of subject areas and is a good starting point for the further study and development of conceptual modeling problems. From the viewpoint of the classification of facilities for the representation of dynamics this approach belongs to the situation-oriented class [5] and makes it possible to construct non-procedural purely declarative conceptual modeling languages. Furthermore, a strict and well developed formal system exists which can be used, with slight modifications and constraints, for a further analysis of this approach, for constructing the abstract syntax and strict semantics of individual declarative conceptual modeling languages, and for developing the logical principles of its application and implementation in an AIS.

It is a question of aletic modal logic [10], the semantics of the possible Kripke universes [11, 12] associated with it, and Khintikka's semantic approach [13, 14]. Actually it is not hard to see that the conceptualization of a subject area in fhe form of an <M, R> pair practically gives us the set theory structure that serves as the area of interpretation of first-order aletic logic systems in the semantics of possible Kripke universes. And this at once makes it possible to use formal languages of aletic logic with modal necessity and possibility operators as the basis of the conceptual modeling language, and to develop a theoretical model approach to the representation of the subject area in knowledge bases.

However, the theoretical proof approach is preferable for the analysis and development of conceptual modeling problems, as demonstrated by Reiter through the example of modeling statics in relational data bases [15]. The ability to design a theoretical proof system as applied to the treatment of subject area statics and dynamics that we have discussed opens the door to the enlistment of Khintikka's semantic approach [13, 14]. On this basis the authors have now completed the design of a theoretical proof approach to the representation and principles of the implementation of situation-oriented conceptual modeling languages, and experimental verification is under way.

Development of Facilities for Formal Representation of Knowledge

The choice of mathematical logic as the principal tool for studying the problems of intelligent AISs makes it possible to outline the following process for the development of facilities for formalization of the representation of knowledge.

Aletic first-order logic, whose interrelationship with knowledge bases was discussed above, can serve as the nucleus of these facilities. The further development of formal facilities for the representation of knowledge can be regarded as a process of step-by-step expansions of this nucleus--which we will call the basic calculus for conceptual schemas--in several directions. In one of them, which does not go beyond the limits of the source system of logic, these expansions will consist in the addition to the basic calculus language of new-non-logical symbols for the representation of required concepts and in the addition to the knowledge base of new non-logical axioms characterizing these concepts. For instance, the enlistment of set theory concepts for conceptualmmodeling can be formally represented as the explicit addition to the language of the knowledge base's nucleus of the new non-logical axioms characterizing these concepts. For instance, the enlistment of set theory concepts for conceptual modeling can be formally represented as the explicit addition to the language of the knowledge base's nucleus of the new non-logical predicate membership symbol and the addition to the knowledge base of axioms characterizing the concept of a set and axiom definitions introducing other set theory concepts (relation, function, etc.). As a result a sort of new calculus-1 is obtained--a formal theory of the first order serving as the nucleus of knowledge bases based on the concept of a set.

Then, as the conceptual design system develops, non-logical symbols can be introduced for representation of the concept of a property or attribute with the appropriate axiomatics and formal definitions of concepts such as essence class and copy-class and genus-species relations, etc. [16]. This will give us calculus-2, which is an expansion of calculus-1 and that makes it possible to represent formally structural relations between subject area essences. Further development of the conceptual system will be reflected in the determination of further calculus-3, -4, etc.

Thus, a "knowledge base calculus" can be constructed by step-by-step expansions of the basis calculus, i.e., a theory serving as a means of formally representing a specific set of general concepts to be used in conceptual modeling.

Further development of the formal system can proceed in the direction of specialization oriented toward classes of subject areas (accounting, diagnosis if diseases, etc.). The isolation of concepts common to a certain class of subject area and their formalization within the framework of a knowledge base calculus will result in the creation of branches from the common formal nucleus (fig 2). The knowledge base for a specific subject area (e.g., management of the production stock of the Elektron Plant), which is what the developer of a specific AIS must design, will include the regular expansion of the calculus for the subject area class, in which all general concepts are linked to this subject area, concepts specific to this subject area are introduced and formally described, etc.

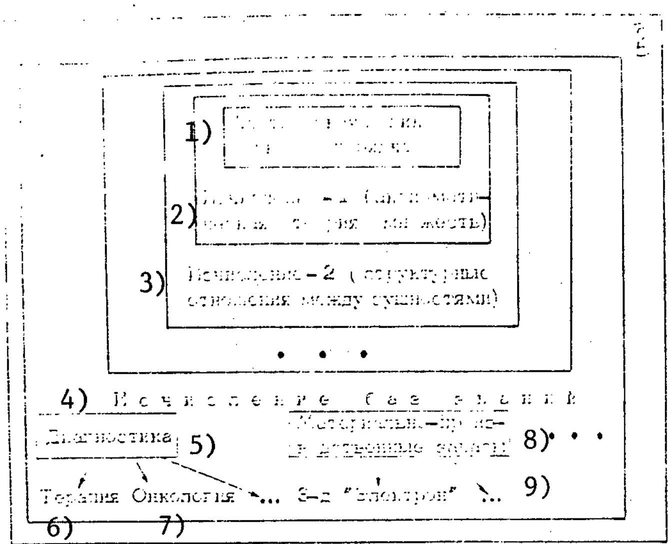


Figure 2.

Key:

1. First-order aletic logic
2. Calculus-1 (axiomatic set theory)
3. Calculus-2 (structural relations between essences)
4. Knowledge base calculus
5. Diagnosis
6. Therapy
7. Oncology
8. Production stock
9. Elektron Plant

Another direction, another "dimension" in the expansion of formal facilities for the representation of knowledge consists in the expansion of logical concepts by going beyond the limits of the system of logic used, i.e., enlisting the knowledge representation of temporal [17], intensional [18], epistemic [19] and other logics. Nested rectangles having branches similar to those depicted in fig 2 can also be obtained in this dimension.

Finally, the third dimension of expansions will make it possible to reexamine such basic concepts as "truth" and "falsehood" and will lead us to multiple-valued, fuzzy and fuzzy-valued logics [20].

Enhancement of Intelligence of AISs

Before the development and implementation of intelligent AISs is it necessary to understand precisely what their intelligence will consist in. AISs have originated, developed and been used outside the field of artificial intelligence. These AISs have had their own achievements and failures and their own, sometimes bitter, experience. How do these AISs--let us call them traditional ones--correspond to and differ from intelligent ones? What are the principles for the development and comparison of intelligent systems? At present the concept of "making AISs intelligent" is rather vague and generally speaking includes several aspects. Let us try to discuss them in a more or less unified context.

Improving the "Literacy" of AISs

The first of these aspects can arbitrarily be called improving the literacy of an AIS or its "level of education." In everyday life we often call a specialist literate in a field if he has a sufficient amount of knowledge accumulated in that field of work. In beginning to work together with him it is not necessary to explain to him the meaning of the basic concepts used in this field and to inform him of well known and easily found rules, facts and recommendations, previously gotten results, etc. The specialist already knows this, having gone through training at school and a VUZ and having gained the required experience in the process of his prior work.

A similar situation can occur also when an AIS having a certain minimum amount of knowledge put into it beforehand plays the role of the partner. In beginning to work with such a system the developer of a knowledge base is spared the necessity of a detailed explanation of the meaning, features and interrelationships of the principal concepts used in conceptual modeling in general and in the subject area class in question (e.g., in the management of production stock) in particular. From the formal point of view this means that the developer of the knowledge base does not himself have to develop and introduce into the AIS the axiomatics for representation of the meaning of general concepts. The required axiomatization has already been done and been introduced into the AIS by the construction of a minimal conceptual schema for this subject area class [5, 8]. In this case it is required of the developer of the knowledge base only to specify the general concepts in his subject area and to represent all the other specific laws, rules and constraints at work in this area, i.e., has has to sort of explain to the system the essential nature and specific features of this subject area and the problems to be solved in it. He can use a fairly broad system of concepts for this purpose without descending to the definition of well known things. All this makes it possible to speak of a definite, at least "superficial," intelligence (literacy) of the AIS. However, there is also an in-depth side of this intelligence.

In ordinary life we tend to distinguish two kinds of levels of education--superficial and in-depth. By a superficial education is usually meant the simple memorization of facts, laws, patterns and methods of operating and any kind of other information, the amount of which can be quite extensive.

However, an in-depth understanding of the essence of the laws and patterns memorized can be lacking. If the person is faced with a real problem which in some way does not fit into well known patterns, he may not be able to use his existing knowledge to find a solution in non-standard situation. A similar picture can occur for an AIS. The system can be rigidly programmed for the representation of knowledge and the solution of problems in some specific class of subject areas. A wide range of situations can be provided for in it; and the software support of various descriptions, queries and changes in the conceptual base have been effectively implemented in it. It will be in a position to "understand" a person for a wide range of questions, but all this will be based solely on the systems programmer's foresight.

The axiomatics of general concepts can be developed in great detail, but their support can be placed deep in the functioning mechanisms of the systems programs. Therefore no change in these axiomatics in association with a change in the general rules and laws of the subject area will be possible without more or less in-depth restructuring of the system itself. The in-depth intelligence of AISs can be spoken of in terms of how simple it is to make changes in them that reflect and support changes in a person's knowledge of his subject area.

The representation of a conceptual schema and conceptual base in the form of axiomatic theories makes it possible to make an in-depth analysis of possible changes in the conceptual base and their consequences. Transactions--programs that are written in a fairly high-level language and are designed for verifying the permissibility of user-requested changes in a conceptual base and for executing these changes--are designed on the basis of such an analysis in traditional AISs and database systems. One indicator of the degree of in-depth "intelligence" of an AIS is how "active" a part it takes in this process: from taking "on faith" transactions written by a person and mechanically executing them to the independent generation of such transactions on the basis of conceptual schema axioms and knowledge put into the AIS concerning the general features of the conceptual base. In the first case changing the conceptual schema requires the redesigning of transactions by a person, and in the second the designing itself and the generation of transactions and all the necessary redesigning and regeneration will be done by the system itself, though perhaps with some prompting by a person.

The next step in enhancing the in-depth education of an AIS involves the capabilities it offers for changing not only the database careated by the designer himself but also that part of the conceptual schema that comes with the system. This is a kind of opportunity to reexamine the knowledge about the class of subject areas toward which the system is oriented. For example, if it is a question of regulation of the contractual obligations of an enterprise or industry it is possible to change the very legislation regulating these obligations. The axioms reflecting the old legislation could be included in the "calculus" of the subject area class and be supplied together with the software, thereby enhancing the system's superficial education. The system will have a high level of in-depth education if it is as simple to modify this calculus to take the new legislation into account as it is to modify the part of the conceptual schema developed by the designer himself that describes the specific subject area.

In summing up the above, let us note that the aspect of enhancement of the intelligence of an AIS discussed here involves expansion of the fund of knowledge (superficial literacy) and the ability to modify knowledge (in-depth literacy) concerning the laws of the subject area. From the viewpoint of the approach discussed in the preceding section enhancement of intelligence has to do with the non-logical expansion of the capabilities of traditional AISs and the flexibility and effectiveness of mechanisms for supporting this expansion.

Knowledge Concerning Knowledge

Another aspect of making AISs intelligent involves their ability to represent and use not only knowledge concerning the subject area, but also knowledge concerning this knowledge. In making a logical analysis of the ability to represent and use information (knowledge) in traditional AISs, it can be said that the descriptive and retrieval mechanisms implemented in them were based on four postulates concerning the capabilities of the developer of the information base. The authors have aribtarily called them the postulates of singularity, infallibility, absoluteness and omniscience of the developer of the information base. The unconditional acceptance of these postulates made it possible to use a simpler, well developed formal system for the representation of knoweldge in an AIS and also simplified the implementation of mechanisms for supporting such information bases.

For instance, singularity postulates unity and a single-meaning for the concepts used, as well as the necessity of consistency in the knowledge base. Any violation of consistency would be interpreted as an error and be rejected by the system.

Infallibility confirms the necessity of consistency and requires unqualified faith in the truth of the AIS's answers, not only because the AIS does not permit errors in its conclusions but, chiefly, because all the knowledge put into it was assumed to be absolutely reliable and true. Thus, if the AIS answered that something is "So," then this would mean that it is "Definitely so," and not "So in the opinion ofthe knowledge base's developer," who, generally speaking can be wrong. Accordingly AISs have not needed mechanisms for separating absolutely reliable knowledge from the opinion of the knowledge base's developer. For the same reason there has been no need to state answers more precisely in the manner indicated above.

Absoluteness means: "Let your word be Yes for Yes, and No for No; whatever goes beyond this, comes of evil." (Matthew, V, 37 [Knox translation]). Accordingly, two-valued logic is sufficient as a formal system for the representation of knowledge.

Finally, omniscience means that the true meaning of any judgment of interest is known. Accordingly, a knowledge base represents a complete theory in which either A or not A is deducible for any closed proposition A.

All four postulates together set once and for all the features of the knowledge about a subject area that would be entered into an AIS. The mechanisms for representing and working with an information base (knowledge base) in traditional AISs were designed under the assumption that the knowledge that can be entered into the system must conform to these postulates. At the same time the knowledge about the very knowledge to be represented (consistency, adequacy, precision, completeness, disregard for the subject of the knowledge) was fixed in these mechanisms as if for all time.

It is easy to see, however, that these postulates portray the developer of a knowledge base as an omnipotent being, at least within the framework of a certain subject area. In other words, the application of traditional AISs is limited to subject areas that have already been studied thoroughly, all the inconsistencies in the assimilation of which have been resolved, and whose basic truths have been established fully. In practice, of course, the degree to which many real subject areas have been fully research can differ considerably from the situation presented above. Even elementary facts can be unknown. For example, the developer of a knowledge base can be informed that products of type p are supplied by either enterprise C_1 or C_2 or by both, but still not know exactly whether C_1 (or C_2) supplies product p. The support mechanisms for information bases in traditional AISs exclude such a situation and accordingly do not make possible the correct representation of work with incomplete knowledge even in such simple cases. However, incomplete knowledge, even the kind presented in the example above, is also knowledge that contains helpful and important information on the subject area, and its loss would be extremely undesirable.

Similarly, the creator of a knowledge base might, in a real-life situation, not have accurate knowledge concerning the truth of a certain judgment; but he might nevertheless be able to make an approximation as to how close to the truth that judgment is. For example, he could say, "The chances are eight out of ten that a product of type p was supplied by enterprise C_1." This is also a kind of knowledge (called fuzzy or imprecise knowledge), which is lost under the postulate of absoluteness.

Rejection of the postulate of infallibility makes it possible to consider the unreliability of knowledge, which is a quite common situation in the real world. And a person, for better or worse, can work and make decisions under conditions of unreliability in the information that he has. Accordingly, the transfer of this capability to an AIS would also signify an increase in its intelligence.

Finally, in real situations a single unified view of a specific problem or specific subject area is often totally lacking. Intead, there exists just a collection of personal opinions, particular incomplete and imprecise points of view, on the basis of which a general integrated view of the core of the matter has still to be developed. The ability to connect up an AIS at this early stage of studying a subject area and use it in the analysis of conflicting opinions for the purpose of grasping a "grain of truth" would signify a considerable growth in the intelligence of the AIS.

In the authors' opinion, on the general level the second aspect of making AISs intelligent consists in the development of systems capable of representing knowledge and working with knowledge that does not satisfy the four postulates. From this perspective a system that can represent incomplete knowledge and work with it (rejection of the postulate of omniscience), for example, must be considered more intelligent than a system designed for the fulfillment of all four postulates. The level of intelligence of a system will be higher the less rigidly its mechanisms are oriented toward a specific configuration of accepted and rejected postulates and the higher its ability to determine independently the characteristics of the knowledge put into it (a kind of self-analysis). Along with this independence would go the ability to select independently the algorithms for receiving an answer to a query and the necessary methods of pinpointing an answer. From the formal point of view, increasing the intelligence of AISs in the respect discussed here must be based on logical expansion of the nucleus of traditional AISs.

For instance, rejection of the postulate of omniscience while keeping the remaining three postulates requires the enlistment of epistemic logic [19, 21]; and rejection of the postulate of absoluteness while keeping the remaining three, the enlistment of several versions of fuzzy logic [20]. Rejection of the postulate of infallibility under the same conditions will require the use of suppostional logics [19]; and rejection of singularity, the use of logics of knowledge and supposition with several subjects [19]. Of course, the rejection of several or all of the postulates would require a combined generalized formal system, which essentially does not exist as of today.

Thus, increasing the intelligence of AISs with respect to the representation of knowledge about knowledge runs into considerable difficulties, above all because of the lack of a reliable theoretical foundation for the design of such systems.

Improving Intelligence Through the Expansion of Methods for Obtaining New Knowledge

The third aspect of making AISs intelligent involves incorporating the ability to simulate methods of reasoning, i.e., methods which a person or system uses to obtain new knowledge based on available knowledge. Traditionally so-called deductive reasoning [4] has long been at the center of attention of logic. In the broad sense, by the deductive method of reasoning was meant methods and rules that guaranteed the truth of a conclusion (new knowledge) given the truth of the premises (source knowledge). In the narrower sense, deduction means finding specific consequences of general propositions (laws). In other words, the rules of deduction make it possible to arrive at or verify the accuracy of consequences from laws already established or postulated by someone.

Generally the goal of our work and the study of a subject area is often not only to draw conclusions from already established laws, but to find these general laws themselves by attempting to formulate a hypothesis and then testing its adequacy. One of the most well known kinds of non-deductive reasoning is so-called inductive generalization, where proceeding from the sum of particular cases and facts an attempt is made to derive a general law [4] which is then verified on the basis of the existence or absence of contradictory deductive consequences resulting from it.

The advancing of hypotheses or the generalization of empirical facts as a form of human intellectual activity plays a very important role in the process of cognition and is probably rightfully considered one of the distinctive features of human intellect. Therefore it is natural to regard as one indicator of the intelligence of an AIS the presence in it of mechanisms for the simulation of non-deductive reasoning--in particular, inductive generalization--and the degree of perfection of these mechanisms. From the formal point of view, the inclusion of various kinds of non-deductive conclusions in a knowledge base represents the expansion of the logical formal system for the representation of knowledge by the introduction of new rules of inference (in place of or in addition to rules of separation and generalization).

Conclusion

This article is devoted to a logical analysis of the intelligence capabilities and problems in the development of so-called intelligent automated information systems. Questions relating to conceptual modeling and the representation of knowledge in these systems are discussed, as well as the role of formal mathematical methods in this process. The initial context for studying problems in the representation of knowledge bases has been formulated and the levels of these problems have been presented. Mathematical logic and the formal methods existing within its framework were used as the principal tool for making the analysis. Using this formal logic as a basis and an approach was developed to conceptual modeling of the statics and dynamics of subject areas. The formal principles for the development of situation-oriented conceptual modeling languages were concisely presented, and a general view was presented of further development of the conceptual system and ways of formalizing it for the purpose of support within the framework of an AIS. Ways and methods of increasing the intelligence of AISs were discussed and a context was created for discussing and comparing their intelligence.

BIBLIOGRAPHY

1. Pospelov, G. S. and Pospelov, D. A. "Artificial Intelligence--Application Systems," MATEMATIKA, KIBERNETIKA, No 9, 1985, 48 pages.

2. Kuzin, L. T. "Models of Artificial Intelligence Systems" in OSNOVY KIBERNETIKI" [Fundamentals of Computer Science], Vol 2, Moscow, Energiya, 1979, pp 409-513.

3. Simons, Dzh. EVM PYATOGO POKOLENIYA: KOMPYUTERY 90-x GODOV [Fifth Generation Computers: Computers of the 90s], Moscow, Finansy i statistika, 1985, 174 pages.

4. Kondakov, I.N. LOGICHESKIY SLOVAR-SPRAVOCHNIK [Logic Dictionary-Handbook], Moscow, Nauka, 1976, 720 pages.

5. ISO TC97/SC5/WG3. Coneepts and Terminology for the Conceptual Schema and Information Base," No 695, 1982, 196 pages.

6. Mendelson, E. VVEDENIYE V MATEMATICHESKUYU LOGIKU [Introduction to Mathematical Logic], Moscow, Nauka, 1971, 320 pages.

7. Steel, T. B., Jr. "Formalization of Conceptual Schemas" in MODELING IN DATA BASE MANAGEMENT SYSTEMS, Amsterdam, North-Holland, 1976, pp 1-47.

8. Yashunin, A. Ye. "Formal Logic Methods of Describing Subject Areas at Conceptual Level" in MAT. METODY V AVTOMATIZIROVANNYKH INFORMATSIONNYKH SISTEMAKH I BANKAKH DANNYKH: SB. NAUCH. TR. [Mathematical Methods in Automated Information Systems and Databases: Collection of Scientific Works], Kiev, IK AN USSR, 1985, pp 31-36.

9. Quine. W. O. MATHEMATICAL LOGIC, Cambridge, Harvard Univ. Press, 1961, 534 pages.

10. Feys, R. MODALNAYA LOGIKA [Modal Logic], Moscow, Nauka, 1974, 520 pages.

11. Kripke, S. A. "Semantic Analysis of Modal Logic. Part 1. Normal Modal Propositional Calculi" in Feys, R. MODALNAYA LOGIKA, Moscow, Nauka, 1974, pp 254-303.

12. Kripke, S. A. "Semantic Discussion of Modal Logic" in SEMANTIKA MODALNYKH I INTENSIONALNYKH LOGIK [Semantics of Modal and Intensional Logics], Moscow, Progress, 1981, pp 27-40.

13. Khintikka, Ya. "Kinds of Modality," IBID., pp 41-59.

14. Khintikka, Ya. "Modality and Quantification," IBID., pp 60-75.

15. Reiter, R. "Towards a Logical Reconstruction of Relational Database Theory" in ON CONCEPTUAL MODELING, New York, Springer Verl., 1984, pp 190-238.

16. Gibbs, S. J. "Conceptual Modeling and Office Information Systems" in OFFICE AUTOMATION, New York, Springer Verl., 1985, pp 193-226.

17. Prayor, A. N. "Temporal Logic and Continuity of Time" in SEMANTIKA MODALNYKH I INTENSIONALNYKH LOGIK, Moscow, Progress, 1981, pp 76-97.

18. Montegyu, R. "Pragmatics and Intensional Logic," IBID., pp 223-253.

19. Kostyuk, V. N. "System of Scientific Knowledge and Its Logical Analysis" in SISTEMNYYE ISSLEDOVANIYA [Systems Research], Moscow, Nauka, 1977, pp 206-223.

20. Pospelov, D. A., editor. NECHETKIYE MNOZHESTVA V MODELYAKH UPRAVLENIYA I ISKUSTVENNOGO INTELLEKTA [Fuzzy Sets in Models of Management and Artificial Intelligence], Moscow, Nauka, 1986, 312 pages.

21. Levesque, H. J. "The Logic of Incomplete Knowledge Bases" in ON CONCEPTUAL MODELING, New York, Springer Verl., 1984, pp 165-186.

8831/9835

UDC 681.3.063

Software Design Algebra

18630204c Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 6, Nov-Dec 87 (manuscript received 9 Jul 87) pp 99-110

[Article by I. V. Velbitskiy under the rubric "General Software for Systems Management"]

[Text] Introduction

The expected improvment in the labor productivity of programmers in the very young and rapidly developing information processing industry has not occurred in the last 20 years. Yet an entire arsenal of modern facilities for the automation of programming has appeared over these same 20 years; it includes languages, compilers, operating systems, software packages, knowledge bases, expert systems, etc. Consequently, the conclusion can be drawn that modern automation facilities are not solving the problems of programming and are not helping programming as a process of of its deadlock. This does not mean that everything done in programming over the 20 years has been bad and of no use; but it just means that everything done has not been united conceptually under a high engineering standard for the development and use of software systems by teams of specialists.

Today questions relating to software engineering are above all questions of a high professional programming standard, a standard for communicating with a new member of our society--the computer. This standard must and will be just as wide-spread as the practice of driving an automobile is today. However, for the moment it is in an uncontrolled manner and is far from perfection. It is developing obviously insufficient for solving today's problems, and a fundamentally new standard for the development (design) of software different from the current customary and traditional approaches is needed for solving the incomparably more complex problems of the future.

For this purpose it is necessary to depart from the customary extensive methods of developing software--from the art of people on their own (chief programmers)--to collective methods and states, the industrial development of software. As the GREAT SOVIET ENCYCLOPEDIA and only industry can exert a decisive influence on the level of development of the productive forces of society.

However, the software industry differs from our traditional ideas of industry in that its essential focus is not the mass production of predesigned products, but rather the design of these products. The actual process of mass producing software is rather trivial, like the process of copying magnetic tapes or cassettes. In other words, the software industry belongs to a non-traditional, creative, intellectual (and not physical) sphere of human activity that results in the design and creation of something new and unique--in this case, software. It is the method, principle, regulated approach to solving each new problem, and the entire plant infrastructure for the development, adoption and implementation of products that are all mass reproduced.

In addition, a program is distinguished from other industrial products by exceptional logical complexity that it contains within a very simple, even elementary, physical form. A program does not wear out; it has no "tolerances and fit" (permissible deviations from the ideally precise dimensions). It absolutely must be precise, however many millions of instructions (analogous to parts in other industrial products) it contains. It is possible to know a program only through an intermediary--via the work of a computer and its environment in their complex interrelationship.

This is the most complex problem that man has had to encounter hitherto in the formation of any industry. Therefore the industrial technology for the development (design) of software must be based solidly on mathematically strict discipline in the organization of work.

The general methodology and algebra for the design of software described in this article can serve as a sufficiently strong foundation for the development of the industrial principles of programming and can ensure their steady continued development in the future.

Graphics Approach to Programming

Problems of software engineering can be solved in various ways. This has been done traditionally by means of an appropriate programming language. The idea behind this approach is simple and natural: If there were a language that satisfied professional requirements, it could be used to write programs easily and simply, they would be understood by everyone, they could be debugged and modified easily. However, such a language has been in the making for more than 25 years now, beginning with ALGOL-60. The "half-life" of each new language is only three or four years. Not too long ago we were talking about Ada, then about the Ada environment, and now about the C/UNIX environment.

There are already more than 2000 different programming languages in existence. This is almost as many as the number of natural languages that have appeared in the world over the entire history of mankind.

Another approach to the development of software engineering is based on the use of graphics and professional symbols in programming. This trend has been reflected in the proceedings of a domestic conference [1] and for two years in a topical issue of a foreign journal [2].

The text (alphanumeric) method of representing information in a computer has served as the basis for programming up the present. It is simple and reliable for hardware implementation; but it is not clear or compact and does not permit full utilization of the advantages of man's associative visual capabilities and thinking. However the capabilities of hardware are now much greater than is necessary for the implementation of the simplest alphanumeric principles for representing information in a computer. Therefore, the thorough use of graphics--charts, tables, figures, diagrams, etc.--is being proposed for the purpose of eliminating these shortcomings.

The idea behind this approach is also simple and natural. People resort to it, for example, any time they need to "design an industry," i.e., to solve the complex problem of coordinating a team of specialists in the production process. First of all various conventional professional symbols and clear and compact graphic images are assigned, and such concepts as illustration, document circulation, and various performance standards are introduced.

The thorough use of graphics in programming makes it possible to approach anew the solution of all the problems of programming, without exception [3], and is well in keeping with the prospects for development of hardware--color displays, large-scale boards, multicolor printers with high-quality reproduction, problem-oriented distributed multiprocessor information processing, etc.

The specific implementations of the graphics approach, or, as they still say, the principles of visual programming, can be quite diverse [2].

Let us discuss the graphics of the R programming method [1, 4-7] in its present-day interpretation.

The R method is a domestic method in which the basic current and very important concepts of programming theory and practice, the graphic form of recording and [words missing] and foreign methods have been successfully combined on a unified conceptual basis. The R method of programming is designed for the development of a broad class of software, including structurally and logically complex software, in all applications of computer technology.

The following constitute the basis of this method:

The industrial concept of the multiloop and multilevel design of software through the step-by-step specification of any non-formal concepts in an algebra of graphic structures. This method makes possible an "assembly" style of programming, multiple applicability and new software store principles.

A unified graphics, not dependent on programming languages, for recording algoirthms, programs, data and processes at all stages of their development.

A system for automating the team's organizational work and making possible the friendly interfacing of specialists (customers, problem formulators, project managers, algorithm specialists and programmers) throughout development of the software project and the "wresting" of it from the developers when the work is done.

GOST [All-Union State Standard] 19.005-85, "YeSPD [Unified System of Project Documentation]. R-Charts of Algorithms and Programs. Conventional Graphic Symbols and Implementation Rules," was developed for the graphics of the R method. This standard does not have prototypes and foreign analogues. It specifies not only the graphic symbols for R-charts and the rules for implementing them and connecting them to one another (graphic structure algebra), but also unified methods for expanding and developing them. R production systems [called RTK] for using the R method have been developed which automate the graphics approach to programming at practically all stages of software development--from formulation of the problem, systems analysis and model design to actual design, debugging, documentation and implementation by the user. RTK systems have been implemented with all domestic broad-application computers and they support the graphics approach to programming in regular operating systems with all the principal programming languages: FORTRAN, PL/1, Pascal, C, assembly languages, etc. (table 1).

Table 1. Specifications for the RTK Support System for the Graphics Approach to Programming

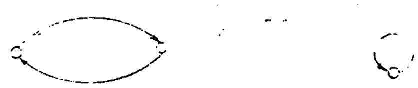
Specification	YeS [Unified System] Computers	SM [Small System] Computers	Personal Computers: DVK-2, DVK-3, YeS 1841 and Elektronika-85
Size of programs:			
Source texts	8.1 MB	1.1 MB	800K
Load modules	4.5 MB	800K	410K
Size of operating documentation	2.1 MB (720 pages)	500K (200 pages)	200K (100 pages)
Minimum hardware configuration:			
Main memory	512K	128K	64K
External storage	5.8 MB	4.8 MB	512K
Operating system	OS 6.1 (OS/370) PDO SVM (CMSVM/370)	OS RV 3.0 (RSX 11-M 4.0)	RAFOS (RT-11) MS-DOS, FODOS PROS
Graphics languages according to GOST 19.005-85	PL/1, Pascal, FORTRAN, C, CHIL, ALGOL-68, RTRAN, assembly language	Pascal, C, FORTRAN, RTRAN, assembly language	Pascal, C, FORTRAN, assembly language

Table 1 (continued)

Supplied on	1 magnetic tape	1 magnetic tape	5 diskettes
Number shipped in 1986	270	200	234

Algebra for Design of Graphic Structures

As is any algebra, the algebra for the design of graphic structures in the R method is defined by a set of elements and a set of permissible operations on them. Two graphic structures are used as its elements and there are three operations: their serial, parallel and nested connection. Both graphic structures have the following form:



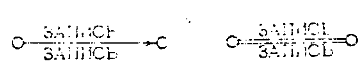
With the use of just horizontal and vertical lines in accordance with GOST 19.005-85:

The same figure represented on standard alphanumeric input/output devices (display screens, printers, etc.) has the form

```
=--->=     =====
|    |     |    |
|<---|     |--->|
```

where two parallel lines or = signs represent two nodes yielding a graph of the loop type illustrated above. The use also of the + sign and of two parentheses side by side, (), representing a so-called special node is permitted according to GOST 19.005-85.

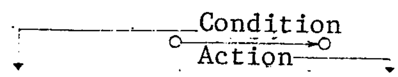
Generally in both these structures there can be as many arcs as desired, in either direction (from left to right and from right to left). Messages of any kind and length can be on the arcs of these structures:



where ZAPIS [MESSAGE] is any text (blank, formula or formal message in a programming language, containing any special characters, tables, figures, etc.) written in one or more lines so that the length of any line of text does not exceed the length of the arc corresponding to the text.

An arc-loaded oriented graph represented by vertical and horizontal lines and consisting of structures (subgraphs) each of which has just one entry and one exit is called an R-chart in accordance with GOST 19.005-85.

Unless specifically stipulated otherwise, messages above an arc in an R-chart specify the condition for passage along the arc; and messages below the arc specify the action to be performed.



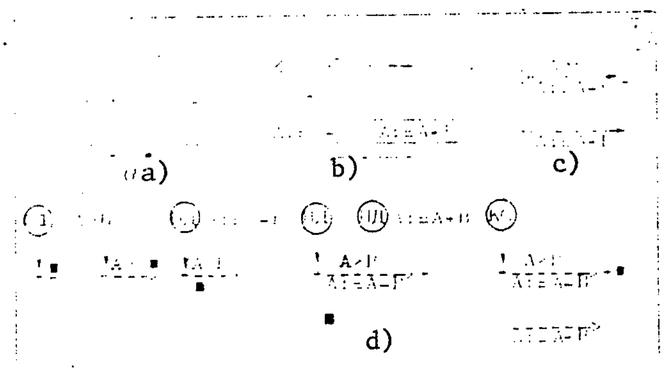
Examples of Condition:

1. $A > 0$
2. $\forall_a \in B$
3. Input of data

Examples of Action:

1. $A := A + 1$
2. $F(X, Y)$
3. Processing of data

It is convenient to represent traditional programming language commands by means of R-charts, whereby the entry of these commands into the computer is carried out 1.5 to 2 times faster (fig 1) and they occupy less memory and fewer display screen lines (fig 2). These advantages make possible larger-module programming with the same hardware, save printer paper, and make the representation of information more compact and legible.



Function Keys:

(1)- Graphic structure of first type
(2)- Move cursor under arc
(3)- Form new arc on screen
(4)- End of structure

Figure 1. Various Forms of Conditional Statement Message: a--traditional text form; b--flowchart; c--R-chart according to GOST 19.005-85; d--protocol for input of R-chart into computer

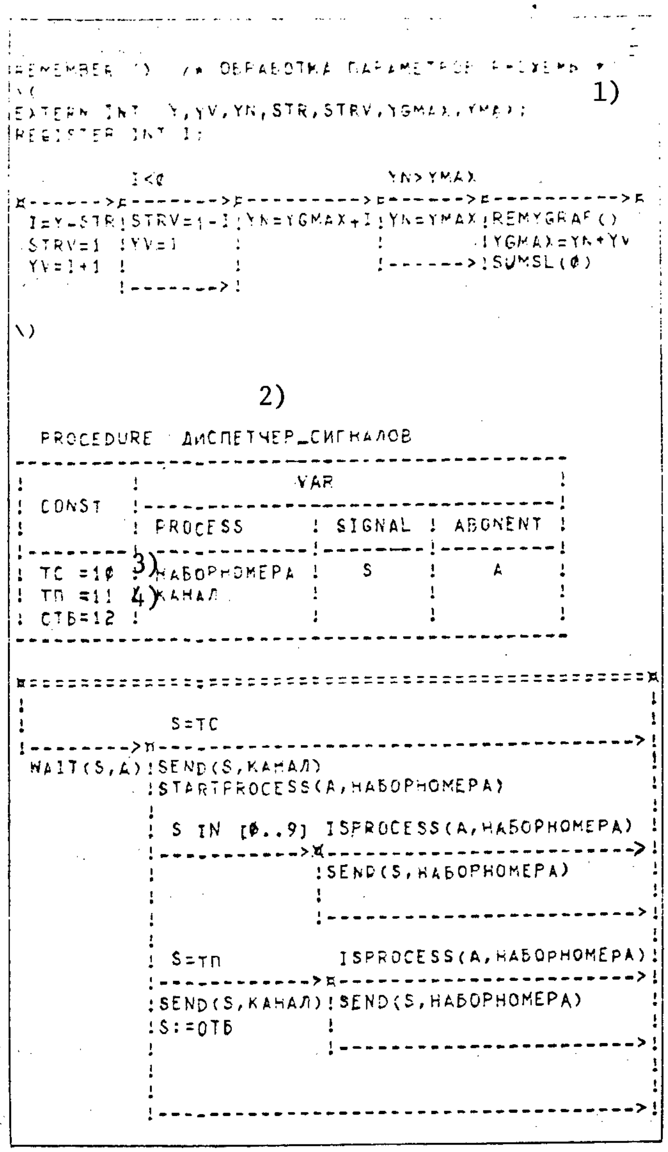


Figure 2. Examples of Program Messages According to GOST 19.005-85 in C (a) and Modula-2 (b) Languages. The graphic approach, as compared with the traditional text approach, occupies fewer bytes of memory and display screen lines, by factors of 1.1 and 1.8, respectively, (for the program presented in C) and of 1.7 and 1.9 (for the program in Modula-2).

Key:

1. Processing of R-chart parameters
2. Signal dispatcher
3. Dialing number
4. Channel

Three operations are designated for joining the two source structures (serial, parallel and nested). They are clearly illustrated in fig 3. As a result of jointing the source structures by means of these operations, new structures are produced, to which all three joining operations can also be applied, and so on without restriction with regard to depth or structures formed (cf. fig 2). Only structured programs are produded as a result of this design. However, in the practice of programming it is necessary to label structures (e.g., for error diagnosis) and to implement jumps to the start or end of structures. Such jumps are called structured and the corresponding commands, structurizers, to distinguish them from the GOTO jump which can be implemented at any point in a program without restriction.

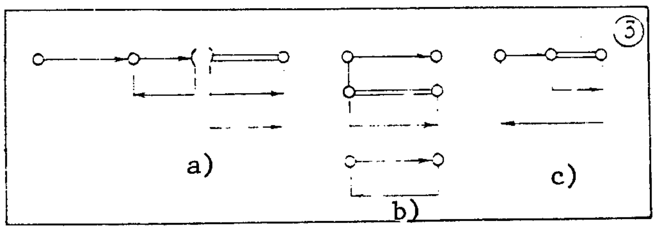
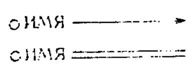


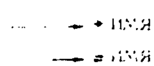
Figure 3. Examples of Serial (a), Parallel (b) and Nested (c) Joining of Structures According to GOST 19.005-85

A structure's label, as specified by GOST 19.005-85, is written inside, near the beginning node of the structure without a space:



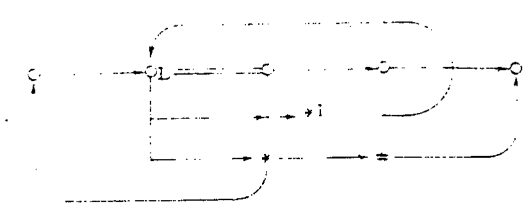
where IMYA [NAME] is a sequence of decimal numbers or an identifier.

The structurizer is written at the end of an arc as IMYA, and before it there is a * or # sign:



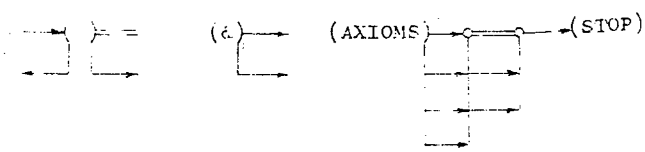
(Structurizer to beginning)
(Structurizer to end)

In the first case a jump is made to the beginning of the structure with the IMYA label, and in the second, to the end of the pertinent structure. In both cases the IMYA label can be absent and a jump is made to the beginning and end of the pertinent R-chart:



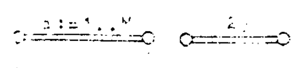
Thus, structurizers make it possible to insert jumps in R-charts without violating the structure of their graphic representation. It is important that the structurizers assign not random jumps, but jumps to the beginning or end of a structure and addressless jumps to the begining or end of the R-chart. Jumps to any arc of a structure, to any condition or operation on an arc, are prohibited and cannot be executed by means of structurizers. Structured programming that is highly effective in describing real-time programs and systems programs, for example, is achieved by the use of this mechanism in the R method. An expansion of permissible operations for the joining of structures in the software design algebra is also achieved by this mechanism. This algebra is capable of the standard expansion of not only permissible operations for the joining of structures, but also the permissible set of algebra elements--the set of permissible graphic structures. Named special nodes and arcs are used for this purpose.

A special node is marked by parentheses, inside of which a sequence of any characters is written, including a blank:



In the first case a special node is used for the separation of two adjacent structures and, accordingly, for the separation and clearer representation of arcs emanating from nodes in different structures. In the second case, specific implementation of the pertinent structure is defined--the parallel execution of arcs emanating from a node. In the third case implementation of a sequence of structures corresponding to the definition of a subject area (axioms, reducers) of a program in the language PROLOG is defined.

A special arc is a double arc without arrows whose inscription defines a particular implementation for the entire structure:

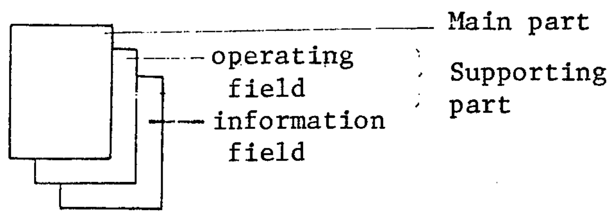


In the first case a loop of the FOR type in Pascal, for example, is defined; and in the second, production arc for a network denoting the time reserve (two weeks) on the route (path) for the completion of a certain job.

R-charts and the algebra associated with them are equally convenient for recording (designing) both algorithms (fig 4) and programs in any language (figs 3 and 5), and data (fig 6) and processes (fig 7). Formal languages and metalanguages are loaded into the graphic form of the R-charts as part of a natural language. The user works at the computer in the graphic algebra of R-charts in a language that is natural for his level (formal or non-formal, as well as in any national language. He gradually adds the details of any non-formal specifications as he writes a precise algorithm in computer language.

Functional Module

In the R method this concept represents a further development of familiar concepts (structure, module, abstract type of data, nowledge base) in the direction of standardization of the concept of an interface. A functional module in the R method is an element of abstraction of the programming process. It consists of a main and supporting part, and the latter includes operating and information fields:



The main part of an R functional module is its external interface, or what a specialist always sees on a display screen or printout when he is running the module.

The operating part of a functional module defines all permissible actions of the specialist in its visible interface part, as well as its communications with the information part and with other modules.

The information part is the module's local knowledge base, or that which determines the specialist's comfort in working with the module: prompts, HLEP texts, teaching texts, certificate data, design dictionaries, documentation forms, deductive axioms and production rules for the local subject area, menus, etc.

A distinctive feature of an R functional module is the language for filling in its parts, or, more precisely, the lack of such a language in the traditional sense of a programming language. All the parts and fields of the module are filled in a natural language in the following forms of representation: structurized text, table, flowchart, chart (more precisely, R-chart) and drawing (Footnote 1).

A text is the most general and least informative, in terms of clarity and speed of recognition of the forms representing information. A text is structured--this means that it has one entry and one exit and is formed according to design algebra rules by means of the operations of serial, parallel and nested joining. In a structured text the entry and exit are formed by means of headings, key words, paragraphs, names of slots and special graphics, e.g.:

□, ◇, Σ□, etc.

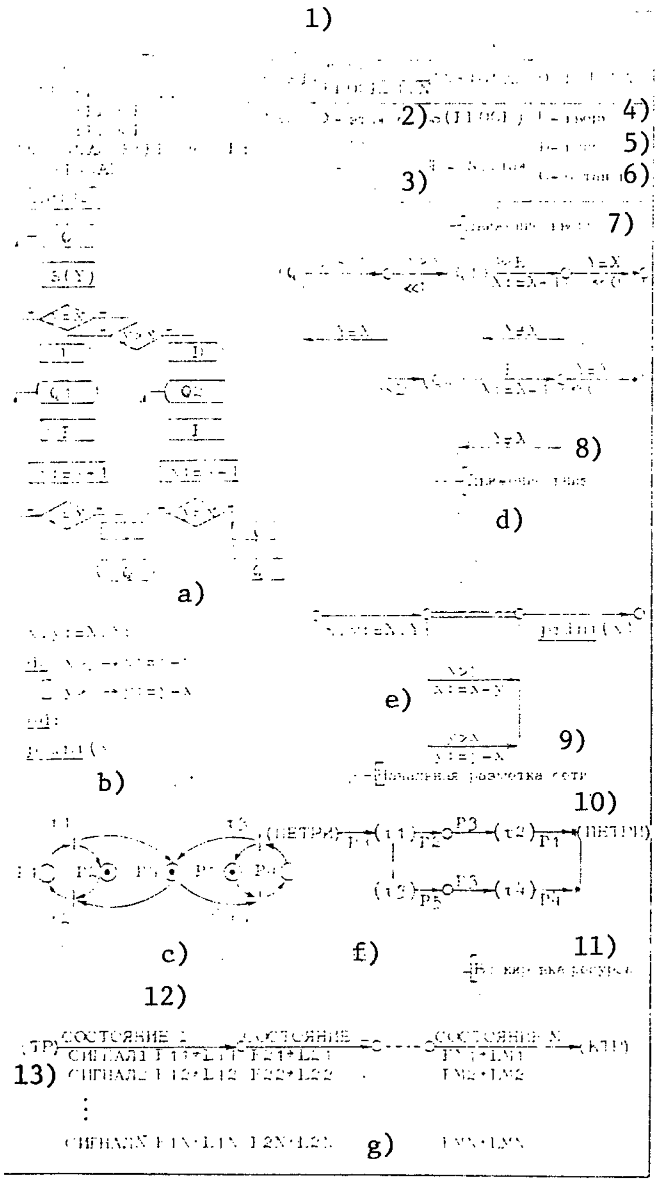


Figure 4. Examples of Writing of Algorithms: a--control of elevator of 20-story building in SDL+ specification language; b--Euclid's algorithm in Dykstra's protected-instruction language; c--classical problem of syncronization of "mutual exclusion" type for use of shared resource (P3) by two parallel processes (P2P1 and P5P4), by means of Petri nets; d, e and f--GOST 19.005-85; g--generalized scheme for writing decision table based on GOST 19.005-85

[Key on following page]

Key:

1. Elevator
2. Floor with elevator
3. Floor of call
4. Up
5. Down
6. Stop
7. Movement up
8. Movement down
9. Marker for beginning of net
10. Petri
11. Resource block
12. State
13. Signal

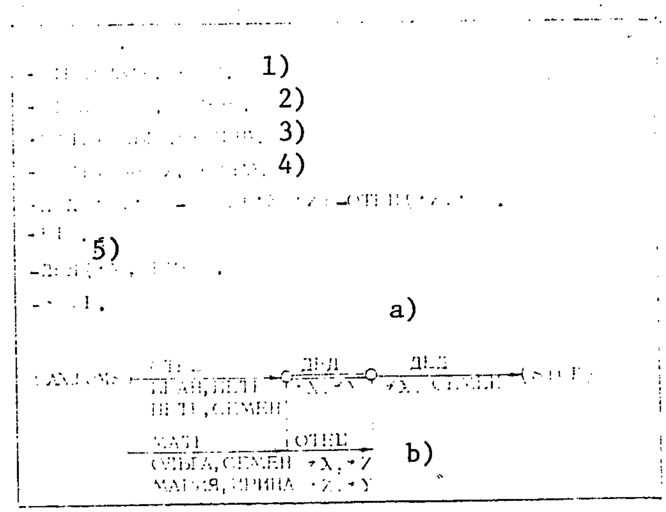


Figure 5. Program for Determining "Who Is Simon's Grandfather?" Written in PROLOG Non-Procedural Language in Traditional Form (a) and Graphic Form in Accordance with GOST 19.005-85 (b)

Key:

1. Father (Ivan, Petr)
2. Father (Petr, Simon)
3. Mother (Olga, Simon)
4. Mother (Mariya, Irina)
5. Grandfather

The formal textual information is separated from non-formal, auxiliary information by means of the appropriate programs in the module's operating field. These same programs link formal information with the appropriate converters (interpreters, etc.).

In all applications the system for standardization of structured text must define, first, the basic set of names for limitation of the text--their syntactical and semantic value--and, second, the minimum dictionary of unambiguously understood terms for a general-purpose definition of the pertinent subject area.

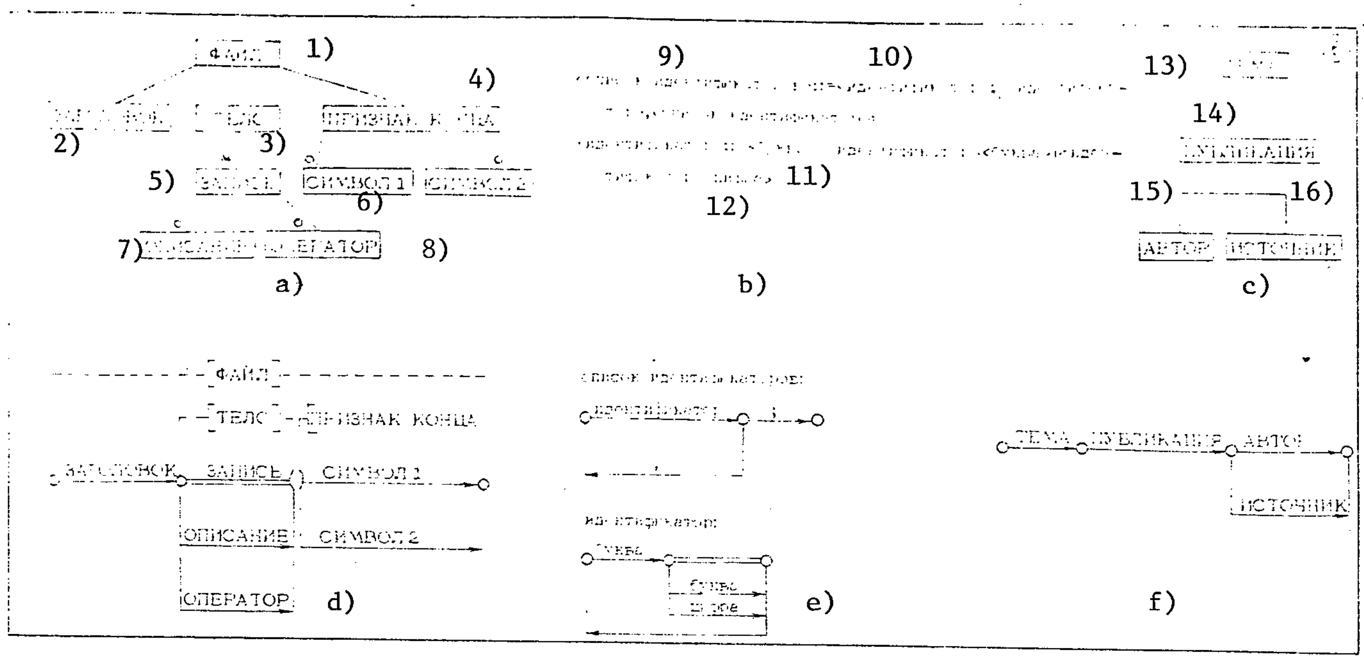


Figure 6. Examples of Writing of Data Structures: According to Jackson (a), in (Bekus-Naurovskaya) Form (b), and Form Used in Data Base Management System (c); d, e and f--Same Data Structures Written According to GOST 19.005-85

Key:

1. File
2. Heading
3. Body
4. End Point
5. Entry
6. Symbol
7. Description
8. Command
9. List of identifiers
10. Identifier
11. Letter
12. Number
13. Topic
14. Publication
15. Author
16. Source

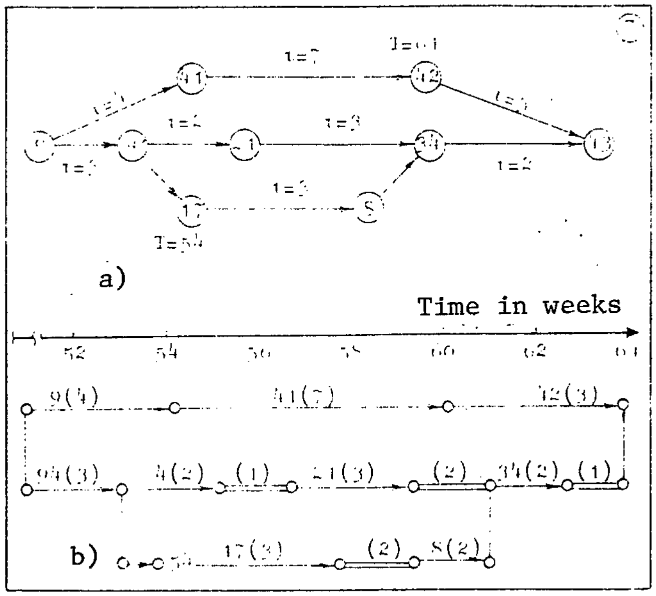


Figure 7. Writing of Networks in Traditional Form (a) and According to GOST 19.005-85 (b). The engineering (double without arrows) arcs denote the time reserve in the completion of jobs. A path in the graph not containing engineering arcs is called a critical path.

A distinctive feature of tables and flowcharts as ways of representing information in an RTM is the unique algebraic principle used in creating them.

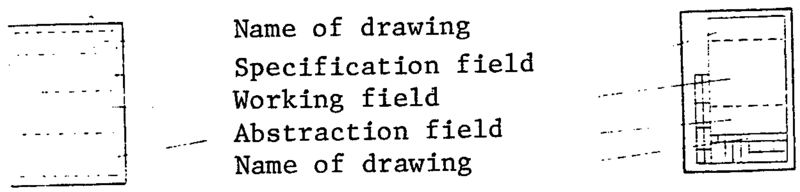
Drawing

The most highly developed and integrated method of representing knowledge, including the methods cited above, is the drawing. It standardizes the description of any subject area, links it to schemes for efficient industrial emplementation, and makes this description absolutely identical in presentation to the contents of the subject area described and the tools used in the process (languages, operating systems, databases, data base management systems, etc.). This kind of standardization makes it possible to improve automation facilities without restriction, including the automation of software interfaces.

A drawing has an official part (frames, labels, mandatory Unified System for Design Documentation and Unified System for Project Documentation inscriptions, etc.) and a content part. The latter has four fields (elements): the name of the drawing, the specifications field, the working field and the abstraction field. The size of each field is arbitrary (a jump to succeeding pages is possible) and the fields can be separated from one another by any method:

Unified System for Project Documentation

Unified System for Design Documentation



The name of the drawing is an arbitrary sequence of characters having in certain instances a short equivalent or a number.

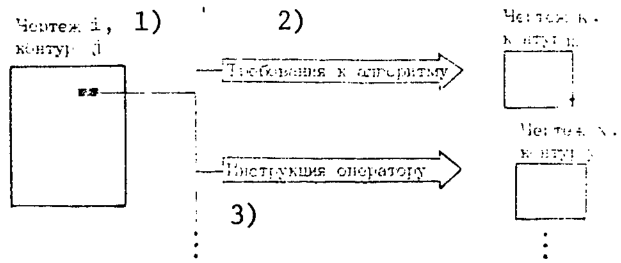
The specification field is a knowledge base of definition of the subject area at the appropriate level of abstraction specified by the drawing. This definition is implemented by means of a structured text (formal or non-formal) with the maximum use of the clearest graphic, formula and other symbols in the pertinent subject area.

The working part is the functional part of the drawing, defining in strict algorithmic form (e.g., in R-charts) the pertinent fragment of the structure to be designed--algorithm, program, data base management system, etc.

Any non-formal symbols used in the drawing's working field or specification field and corresponding to mathematical texts after the word "where" are defined in the abstraction field.

Drawings (more precisely, their content part) can be linked with one another hierarchically via the device of display of any symbol or concept on the drawing. A concept can be displayed in the abstraction field of a drawing, as well as by means of a drawing at the next level or the same level of the hierarchy--a variant definition (fig 8). Concepts displayed on a drawing are highlighted in some way (by color, shade, etc.). A hierarchy of drawings can be multileveled and multidimensional.

Besides the hierarchy, drawings can be united into loops the link between which is not strict or algorithmic but rather informational: "At a certain place in the next loop something is said about this" or "the requirements for this algorithm are formulated at a certain place in the engineering assignment loop," etc. A project's loops can be quite diverse; e.g., the customer's information loop (engineering assignment), and simulation, algorithmic, working, work time optimization, and operating documentation loops. The link between concepts in various loops can be branching and named (by means of a menu); for example:



Key:

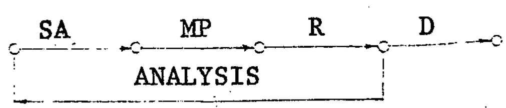
1. Drawing i, loop j
2. Requirements for algorithm
3. Instructions to operator

This system of branched links between the various concepts in drawings (with an electronic, paperless system for implementing it) makes it possible to efficiently form a flexible interface for the thorough and rapid understanding of the project.

Production Process

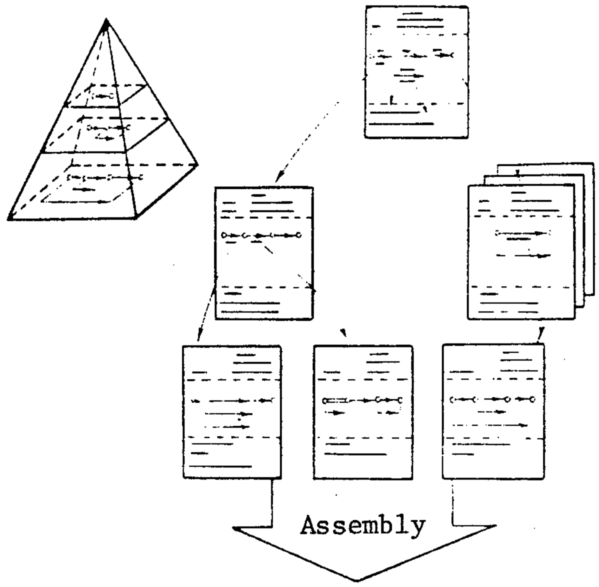
In the R method the production process is established at the stage of engineering preparation for work from R functional modules programmed for that purpose. Particular flexibility of the process is achieved by this aspect of the R method. It can be tailored to the specific conditions of a specific team, thus enabling the planned improvement of the quality of software and the labor productivity of programmers.

The concept of a universal production cycle is used in forming a specific production line in the R method. This means that any development (and/or production or use) of a software system must have the following universal production cycle for its execution:



where SA stands for systems analysis, MP for model designing, R for the work itself, D for documentation of the work, and ANALYSIS for the iterative process of improving the work done.

These names are arbitrary and are replaced by equivalent ones at other stages and steps of the production process, such as: predesign research; draft, detailed, and working design; prototype development; expertise of design; debugging; testing, job completion report; monitoring; production; and support.



13
5417173.00054-01 1 01

Calculation of Machine Time Expenditure

----Specification Field----

Purpose:	Calculation upon request of machine time expenditure and printout of results.
Input:	Machine time expenditure database.
Output:	Machine time expenditure table.
Algorithm:	Request name of database and open it; if database not found, request repeated; request data regarding form of calculation; printout of results in form of table.

----Working Field----

Description of variables.

```
#------------------------>#--------------------------------------------->#
|Request name of database|   Request regarding form of calculation.
|                        |   Calculation and printout of table.
|Database not found      |
|<-----------------------|
```

[Continued on following page]

----Abstraction Field----

E	Description of variables: VAR U :DEFREL; LM:ALFA;	E	Request name of database: Write ("Name of Database"); READLN(NN);
Ch	Database not found		
Ch	Request regarding form of calculation		
Ch	Calculation and printout of table		

Figure 8. Make-up of Drawing Based on GOST 19.005-85 and Chart for Programming by Means of Drawings

The development of programs under the R method is done according to a preprogrammed route. As the result of multiloop and multilevel designing, a set of software system drawings is created which can have alternative (non-working) branches that test various implementation ideas, as well as extra branches for possible generation depending on where the design is used. Design development includes the stage of the assembly of drawings for the purpose of forming a deliverable version of the product or an experimental form (prototype) for subsequent regeneration at the production stage.

In the process of development the software design is tested (beginning with the earliest stages of non-formal design), in particular, for conformity to the project's documents and specifications. Appropriate automated procedures have been developed for this, which are based on the R method's clear graphic form of assigning algorithms, formalized methods of estimating the complexity of software, and automated test generation.

The documentation stage in its traditional sense is absent in the R method, for the set of software design drawings as described in GOST 19.005-85, which corresponds to the "Program Text" document of the Unified System of Project Documentation, completely replaces the "Program Description" document. And the design's user documentation is produced not from scratch as is usually the case, but rather by editing and automated generation of the appropriate set of drawings from the specification field. Its volume is smaller than that of traditional documentation, since it contains only the basic information most necessary for using the system, and the user is referred to the appropriate set of design documentation drawings for any further information. The design and user documentation is created by unified production operations that contain the appropriate forms and templates, for ease of comprehension and mastery.

An edited design tree (set of drawings), instructions for generation of the system, a test program with a control test generator and a set of user's documentation are included in the set of software supplied by the R method.

The production of software by the R method involves not only the repeated production of the software, but also a large set of operations for tailoring it to specific conditions of use. During production the original software is regarded only as a prototype and it is actually redesigned on the macro level so as to make the final software product more convenient and efficient. A number of measures relating to marketing, the gathering of statistics during use of the design, the analysis of complaints and modification of the design as the result of experience gained in using it, are also involved in production. All this work is done independently of the developers of the design on the same equipment as development itself and guided by routes and R functional modules specially programmed for the production process.

Software produced by the R method can be used completely independent of the designer and producer. The user of a software system developed by the R method can, using the documentation, investigate as deeply as he wishes the system's operating principles and, depending on his experience, modify it himself for request that modifications be made by the support or production services.

Organization of the team in the R method is performed through a paperless, interactive electronic mail network. In this network all specialists (customers, functional specialists, software developers and encoders, project administrators, etc.) are arranged in a hierarchy. The work of each is predesignated by the process route entered into the computer at the preparation stage for each type of specialist and each stage of work performed by them. Each specialist in this system has a log, on which are indicated his last name, code (sign-out number), data access password, the list of his subordinates, a list of current, planned, and completed work, etc. The computer sees to it that the specialists interact with it according to protocol and blocks any unauthorized access.

In the R method each job also has a log, on which are indicated its name, registration number (code) and organizational and technical data. The organizational data include information concerning how the job was stated, when and to whom it was assigned, when and how it should end, to whom and in what form it is being presented, etc. These data are transferred to the system with a non-formal specification in the abstraction field of the pertinent project drawing. A job completion chart is formed in the system based on these data and observance of this schedule is automatically monitored. Information concerning the production route for the job, i.e., the sequence of operations with the pertinent organizational data for each, is communicated in the technical part of the log. The majority of the data in the organizational and technical parts of the log for a job are produced automatically.

The interchange of information in the interactive electronic mail network is performed by means of messages, which are stored by the system for each user in a previous communication session and are read out to him when he accesses the system through his code and password. The messages are of three kinds: assignment messages--from managers to their subordinates; system--when the system automatically reminds the user of preplanned deadlines for the completion of an individual job or announces failure to meet a deadline; and other messages--from maintenance services, project administrators, between programmers, etc. Each type of message has its own processing procedure, fixed in the system and monitored during the production line's operation.

Comparative Analysis

An R-chart is an arc-loaded graph. There are two functionally complete and adequate methods of representing a graph: a node-loaded graph and an arc-loaded graph (e.g. Moore and Mealy automata, flowcharts, network,graphs, etc.).

The node-loaded graph has historically been most widely used in programming--flowcharts, SDL diagrams, block diagrams, ISO standard diagrams, etc. However, such a graph has two shortcomings. It is not supported in a computer (it cannot be used as a program for input to a computer) and writing by means of these graphs, though clear, is not compact. It is not convenient to reproduce it on current input/output devices (displays, printers). In compactness of writing it does not compare favorably to programs in traditional procedure-oriented languages. A method in which two character systems are used (graphic at the precomputer design stage, and traditional line for input and working in the computer) proved in practice to be complex and inefficient. Therefore, a node-loaded graph is used in programming only in exceptional cases for particularly important projects as a means for the manual designing and documenting of software.

GOST 19.005-85 and the R method based on it use an arc-loaded graph. The two methods of representing a graph are compared in figs 1, 4, 6 and 7; and R-charts based on GOST 19.005-85, using the idea of an arc-loaded graph, and various national standards (USA, Great Britain, Japan, FRG and the Netherlands) using a node-loaded graph are compared in table 2. The following conclusions can be drawn from an analysis of the illustrations presented.

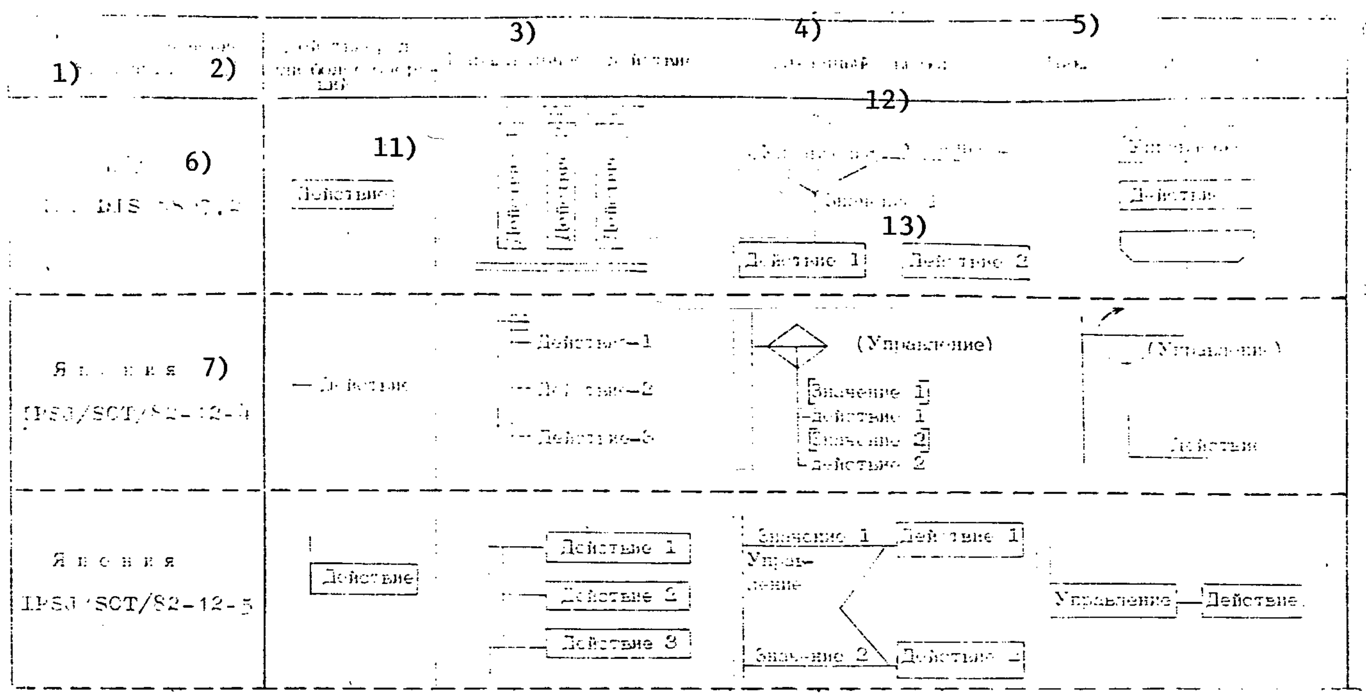
First, R-charts, while not inferior to flowcharts (the short and general term we will use for figures based on the node-loaded graph) in terms of clear representation, are more compact (three to four times on average in terms of alphanumeric representation). However, flowcharts are more customary.

Second, R-charts, having only horizontal and vertical lines, are better (more simply and efficiently) implemented in present-day computer input-output devices (raster and alphanumeric).

Third, GOST 19.005-85 (because of the particular informational compactness of the R-charts on which it is based) not only provides a graphic image and the principal specialized symbols necessary for a cohesive description of algorithms, programs, data and processes. It also provides a standard universal method of expanding (developing) them and the algebra for designing R-charts from very simple elements, both basic ones and those produced as the result of their expansion; and it introduces into programming the concept of the drawing as the next, higher level of abstraction of a structure.

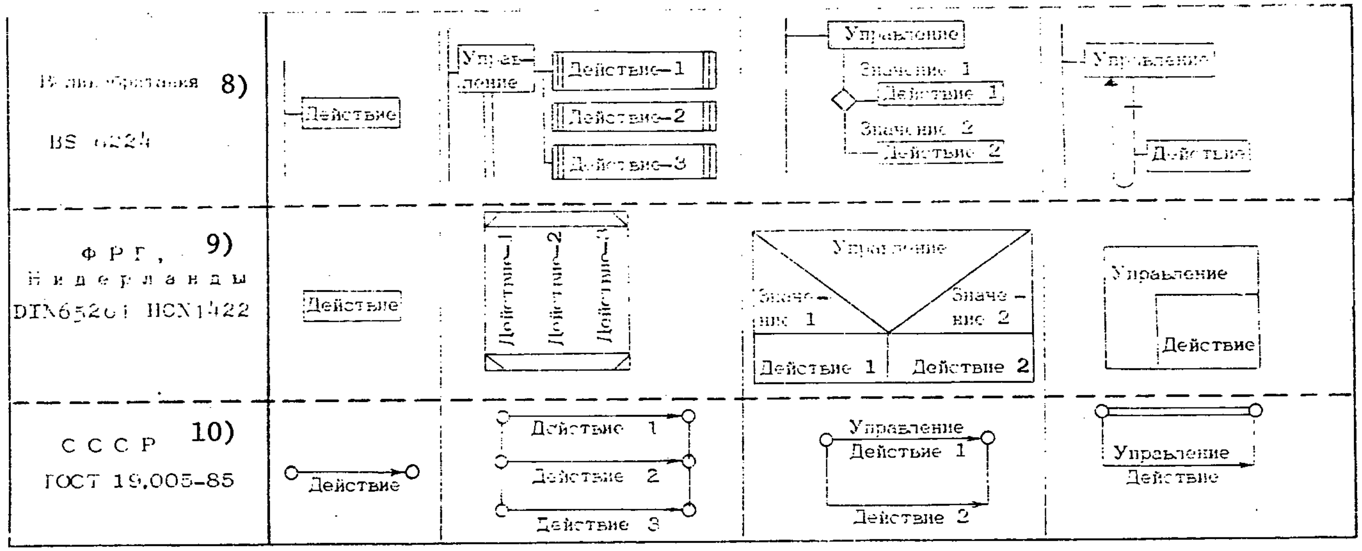
The graphic style, and especially the R-chart algebra described above, opens up new possibilities for the international unification and standardization of programming languages, while at the same time ensuring continuity with their level of development. In fact, when languages are loaded using graphic representation e.g., R-charts based on GOST 19.005-85, the set of keywords of

Table 2. Examples of Specification of Graphic Structures in Various National Standards



[Key and continuation on following page]

Table 2. (cont.)



Key:

1. Country and name of standard
2. Action, one or more operations
3. Parallel action
4. Double option
5. Loop with precondition
6. USA
7. Japan
8. Great Britain
9. FRG, Netherlands
10. USSR
11. Action
12. Control
13. Value

the IF, THEN, CASE, WHILE, type is eliminated and dependence on individual national alphabets is eliminated. The language becomes international--graphic, pictorial and clear.

The functional basis of the languages also becomes structurized. It is reduced to three concepts: the logic expression (written above the arc of an R-chart), and the assignment command and the procedure command, which are written below the arc. All three concepts have identical application semantics in present-day languages and assimilar description syntax. Therefore, the graphics approach erases the differences between languages by making these differences nonfundamental.

The graphics approach as defined in GOST 19.005-85 makes it possible to use any non-formal symbols based on the appropriate basic graphic elements. This in turn makes it possible to automate not only the programming process, but also the software design process, and the programming language in the present-day approach occupies a very small part (about 10 percent) of this process. In other words, the programming language's role is not decisive over the entire range of problem solving with the computer. The language itself (any language) is loaded into (or is included as a part of) the non-formal notation system and is used essentially only at the program coding stage [6].

Conclusion

Experience is the test of any truth. Experience in using the R method has a longer than 10-year history. The R method went through setbacks during its first implemenations, repeated improvements in the implemenation of principles and the principles of implementation on the most diverse computer equipment, the "qualified" comments of "Aesop's foxes," and the throes of standardization and international approval.

This experience is reflected in more than 600 publications relating in one way or another to the R method, in the proceedings of annual (since 1978) training seminars, as well as in the proceedings and decision of its first all-Union conference (1983)--the only conference ever devoted to a single domestic programming method. In the proceedings of this conference an entire volume ("Ch. 2. Opyt primeneniya" [Part 2. Experience of Using], 203 pages) was devoted to a users' discussion in thesis form of the merits and shortcomings of the R method, the huge improvement in labor productivity, its practically unlimited areas of application, and the new high programming standard that is making possible an excellent strategy for the development of domestic software design. Over the past three years developers and industry and state software suppliers have made over 2000 shipments of RTK production systems (cf. table 1) to a wide variety of organizations. Nevertheless, in order to answer the rhetorical question "Well, what of it?" it is necessary to try to write one's own program using its principles.

At first you will experience natural difficulties, since you will have to think not as you have become accustomed to, as you were taught. But then three truths will be revealed to you; First, you will understand where the impasse lies in the traditional approach to program development. Second, you will see the unlimited capabilities of this new method and of the application of your own efforts for developing it. Third, you will understand its uncommonly universal character and how it makes possible interface with other methods and other ways of thinking.

Footnotes

1. Obviously, the concept of "natural language" can include any professional language (e.g., the language of physics, mathematics, programming). A professional lnaguage can be formal (Pascal, PROLOG, etc.) or non-formal (predicate calculi, the writing of differential equations, etc.), strict (in mathematics, physics, etc.) or non-strict (in medicine, biology, etc.).

BIBLIOGRAPHY

1. R-TEKHNOLOGIYA PROGRAMMIROVANIYA: TEZ. DOKL. 1-Y VSESOYUZ. KONF. [R Programming Method: Theses of Papers of First All-Union Conference], Kiev, IK AN USSR, 1983, 349 pages. (Part 1. "Sredstva avtomatizatsii" [Automation Facilities], 116 pages; Part 2. "Opyt primeneniya" [Experience in Applications], 203 pages; Part 3. "Tekhnologiya programmirovaniya mikroprotsessornoy tekhniki" [Method of Programming Microprocessing Hardware], 30 pages).

2. COMPUTER, Vol 18, No 8, 1985, 94 pages.

3. Yershov, A. P. "Personal Computers--Ancestor of Mammals in the Dinosaur World of Collective-Use Computer Centers" in "Dialog-82-mikro. Materialy vsesoyuz. kong." ["Dialog-82-mikro". Materials of All-Union Conference], Pushchino, ONTI NTsBI AN SSSR, 1983, pp 9-25.

4. Glushkov, V. M. and Velbitskiy, I. V. "Programming Methods and Problems in Automating Them," USIM, No 6, 1976, pp 75-93.

5. Mikhalevich, V. S. and Velbitskiy, I. V. "Informatika i promyshlennaya tekhnologiya programmirovaniya" [Enformation Science and Industrial Programming Methods], Kiev, 1984, 33 pages. (Preprint, Institute of Computer Science, Ukrainian SSR Academy of Sciences, No 84-58.)

6. Velbitskiy, I. V. TEKHNOLOGIYA PROGRAMMIROVANIYA [Programming Methods], Kiev, Tekhnika, 1984, 279 pages.

7. Velbitskiy, I. V. and Lavrik, V. N., editors. INFORMATIKA. R-TEKHNOLOGIYA PROGRAMMIROVANIYA I OBUCHENIYE: SB. NAUCH.-METOD. MATERIALOV [Information Science, R Method of Programming and Teaching: Collection of Scientific Methodology Materials], Odessa, Pedagogical Institute, 1986, 159 pages. (Part 1. "Algebra programmirovaniya i problemy obucheniya" [Programming Algebra and Problems in Teaching], 41 pages; Part 2. "Tekhnologicheskiye kompleksy programmistov kak instrumentalnaya baza obucheniya" [Programmer's

Production Systems as Teaching Tool Base], 39 pages; Part 3. "Obucheniye v R-tekhnologii programmirovaniya" [Teaching in R Programming Method], 29 pages; Part 4. "Ispolzovaniye R-tekhnologii v nauke i promyshlennosti" [Use of R Method in Science and Industry], 50 pages).

8831/9835

UDC 621.3:681.3

Principles in the Creation of Electrodynamics Knowledge Base

18630205 Moscow UPRAVLYAYUSHCHIYE SISTEMY I MASHINIY in Russian No 6, Nov-Dec 87 pp 92-98

[Article by O. V. Tozoni]

[Text] General topics and design principles of subject-specific knowledge bases for automated research and design systems for developing equipment based on new technology are examined in work [1]. The use of these principles to create knowledge bases devoted to electrodynamics will be examined here. A subject-specific knowledge base should contain a set of mathematical models that have been constructed by analytical transformation of a single theoretical model, the theoretical model itself, and chains of analytical modules connecting the theoretical model with each mathematical model. For electrodynamics, this theoretical model is Maxwell's equation. The purpose of creating an electrodynamics knowledge base is to provide reliable computer simulation of the electromagnetic processes involved in different designs for new electrical and radio engineering devices.

In the initial design stage, the structure of such a base may be represented as a tree whose top is one of Maxwell's equation and whose branches are chains of analytical and numerical modules (algorithms for the individual stages in the transformation of source information). Each open-bottom chain of the sequentially connected branches of the tree forms a fragment. This fragment represents a numerical-analytical method of calculating (simulating) the electromagnetic processes in a specified class of designs. In the first stage the upper, analytical part of each fragment fulfills a solely instructional function and is part of the "explanation block," which makes it easier to understand the electromagnetic process under study. The lower, numerical part of each fragment comprises the contents of the mathematical model (the lagorithms for its mathematical analysis).

It will be recalled that the specified base structure is built on the following considerations:

the theoretical model, which is supplemented with specific boundary conditions, contains in hidden form all the information about the electromagnetic process involved in each specific electromagnetic device;

the calculation (computer simulation) of the electromagnetic process is equivalent to the transformation of the information contained in the theoretical model and the boundary conditions from a hidden form to a visible form that a specialist can use in decision making;

in the calculation of an electromagnetic process the amount of information converted from hidden to visible form is always limited;

because of the methodological error associated with the difference approximation of multidimensional differential operators, none of the existing numerical methods can guarantee a magnitude of simulation error that would ensure the reliable operation of automated research and design systems;

the methodological error of modern numerical methods, even greather than the desired nonuniform distribution [sic]; on the other hand, specialists designing devices try to provide the maximum possible electromagnetic field intensity in the smallest possible size device, which in turn results in a drastic spatial nonuniformity of the field intensity distributions;

if a portion of the transformations of the theoretical model are done using analytical methods, fewer transformations by numerical approximation will be needed to obtain the end result. This in turn will reduce the computer simulation error. It is therefore necessary to combine numerical transformations with analytical ones and to use mathematical models in which the number of exact transformations of equations in the theoretical model is maximized and the number of approximate numerical transformations done by computer is kept to a minimum;

modern numerical analysis methods make it possible to adapt computers for the reliable solution of systems of ordinary differential equations and (with less precision) to solve integral equations relative to a function in two variables; therefore, analytical transformations of the theoretical model should be directed toward decreasing the dimensionality of the mathematical model, and the number of variables in the functions that are subject to numerical transformations in the mathematical model should not exceed two.

Each analytical transformation of Maxwell's equations requires extensive theoretical research; however, it is sufficient to do this once and then use the result to model a set of different electromagnetic processes. The ultimate purpose of the analytical transformations is to represent the desired distribution of the electromagnetic field in the form of compact formulas containing combinations of tabulated mathematical functions and algorithms for actions on them. This goal is not always attainable, and it is often necessary to limit oneself to discovering additional relationships that make it possible to replace the theoretical model (Maxwell's equation) by equations with a lesser dimensionality. As a rule, this type of analytical transformation exceeds the precision of the mathematical model, shortens calculation time, and permits an estimation of the methodological error.

We will try to classify the analytical transformations of the theoretical model and show a technique for using them to create effective mathematical models.

Methods of creating mathematical models.

1. Narrowing the class of electromagnetic processes by accepted assumptions that simplify the equations of the theoretical model and use general analytical transformations that are invariant relative to the form of the components of a design. If the rate at which the field's sources change over time is small or equal to zero, the field is a quasistationary or stationary (static) field that breaks down into an electrostatic and a magnetic field. In this case general transformations of the secondary source method [2] may be used on a theoretical model simplified by the additional condition $\partial D/\partial t = 0$. These secondary sources reduce the dimensionality of the mathematical model and the area in which the unknown distributions are sought. If the sources change over time in a sinusoidal manner, the transition from instantaneous values to complex amplitudes excludes the time dependence, thereby reducing the dimensionality of the mathematical model to unity.

2. Narrowing the class of designs of electromagnetic devices by the characteristic features of the forms of the components for which existing analytical transformation methods may be used. If a design's components are in the shape of long sections of parallel cylinders, the field is plane-parallel, the dimensionality of the mathematical model is reduced, and the method of the theory of the complex variable functions is used. If the components have the form of coaxial bodies of revolution, the field is plane-meridianal and the dimensionality of the mathematical model is reduced. If a long electromagnetic system is connected by thin conductors, electrodynamic circuit methods are used.

3. New analytical techniques that have been specially developed for a specified narrow class of designs for electromagnetic devices, for example, a toothed magnetic system [3], moving conductive band [4], etc.

Mathematical models of the secondary source method. The secondary source method, whose essential features are described in the following [2], is the most general analytical transformation of Maxwell's equations suitable for computer simulation of electromagnetic processes in a wide class of designs for electromagnetic devices.

When calculating the field in an electronic device, the electromagnetic system may be considered a heterogeneous medium that has been introduced into an electromagnetic field that is in a vacuum and has been created by the given primary sources. From Helmholtz's decomposition theorem [5] it follows that the effect of this heterogeneous medium on the electromagnetic field is equivalent to the presence of secondary sources--charges with the density ρ and currents with the density ∂. If the distribution of all of the secondary sources occurring in the electromagnetic system are found, it is easy to determine the entire field inasmuch as the problem is reduced to one of computing the electromagnetic field in a vacuum based on the known distribution of all of the sources.

For any distribution of sources in an unbounded vacuum, a quasistationary electromagnetic field is determined by the following formulas:

$$\mathbf{B}(Q) = -\frac{\mu_0}{4\pi} \int\limits_V \frac{\delta(M) \times r_{QM}}{r^3_{QM}}\, dV_M,$$

$$\mathbf{E}(Q) = -\frac{1}{4\pi\varepsilon_0} \int\limits_V \rho(M) \frac{\mathbf{r}_{QM}}{r^3_{QM}}\, dV_M - \qquad (1)$$

$$-\frac{\mu_0}{4\pi} \int\limits_V \frac{\partial \delta(M)}{\partial t} \cdot \frac{dV_M}{r_{QM}}.$$

There is a problem when computing the electromagnetic field in a partly-heterogeneous medium--in the space of a device whose components' materials have different characteristics $\varepsilon(Q)$ and $\mu(Q)$, where $Q(x_Q, y_Q, z_Q)$ are coordinates of their points in space. To solve this problem, work [2] proposes transforming Maxwell's equations for a heterogeneous medium to the form of equations for an unbounded vacuum. When this is done, the effect of the heterogeneity of the medium is taken into account by the distribution of the secondary sources, the calculation of which is reduced to the solution of integral equations.

In the case of an unbounded vacuum, to create a field of the vectors B and E that is identical to the field in a heterogeneous nonconductive medium it is necessary and sufficient to do the following [2]:

replace the primary field sources, i.e., replace the current ∂ by $\mu\partial$ and the charge ρ by ρ/ε;

introduce secondary sources, i.e., magnetization currents with density

$$\delta_M = -\frac{1}{\mu_0} \cdot \frac{\mathbf{B} \times \nabla\mu}{\mu}, \qquad (2)$$

and polarization charges with volume density

$$\rho_p = -\varepsilon_0 \cdot \frac{\nabla\varepsilon}{\varepsilon}, \qquad (3)$$

whose effect for the field of the vectors B and E is equivalent to the effect of a heterogeneous medium.

At the media's interface S, the characteristics μ and ε change abruptly, their gradients are indefinitely large, and consequently their densities ∂_M and ρ_P are indefinitely large. The effect of the interfaces is therefore estimated by the surface densities of the secondary current (j) and charge (σ) sources in accordance with the formulas

$$\mathbf{j} = \frac{2}{\mu_0} \lambda_M \mathbf{B} \times \mathbf{n}_Q , \quad \sigma = 2\varepsilon_0 \lambda_n \mathbf{E} \times \mathbf{n}_Q , \tag{4}$$

where

$$\lambda_M = \frac{\mu_i - \mu_e}{\mu_i + \mu_e} , \ \lambda_p = \frac{\varepsilon_i - \varepsilon_e}{\varepsilon_i + \varepsilon_e}$$

and the indexes i and e correspond to the inner and outer sides of the interface S.

By substituting (1) into (2) through (4), we obtain integral equations relative to the distributions of the secondary sources that we are seeking.

Obtaining a direct solution of the integral equations in the mathematical model that has been derived in this manner is difficult in many cases that are important from a practical standpoint. The equations are incorrect in the sense that the small error allowed in calculating the right-hand portion may result in as large an error as desired in the secondary-source density that is being solved for. To regularize the new mathematical model, work [2] proposes the technique of introducing additional information about the integral properties of the secondary sources into the nucleus of the integral equation and shows that such information is easily discovered from the theoretical model in each specific problem. The value of the mathematical model is determined by the volume and quality of the information used in creating it. The more additional information about the electromagnetic process that is included in the mathematical model, the lesser the amount of calculations that are necessary when solving a problem on a computer. The more precise the information, the higher its quality and the more precise the results of the numerical solution. Introducing additional information into the mathematical model narrows the class of functions in which the solution is sought and reduces the number of calculations.

From the property of duality in electrodynamics [6] it follows that different distributions of vector or scalar quantities may be sources of one and the same field. The dimensionality of the mathematical model should therefore be reduced by using (only where possible) scalar secondary sources.

Work [2] presents examples of different mathematical models that have been derived on the basis of the secondary source method and that represent sytems of second-order integral Fredholm equations, for example

$$\sigma(Q) + \int_S \sigma(M) K(QM) dS_M = f(Q), \tag{5}$$

where K(QM), the nucleus of the equation, and f(Q), its right-hand part, are known functions that depend on the distribution of a device's primary sources and geometry.

If the medium is part-conductive and its primary sources are sinusoidal currents, the dimensionality of the mathematical model increases substantially. Optimizing the model by using methods based on the theory of potential made it possible to represent the model in the form of a system of integral equations having a minimal dimensionality [6] relative to the complex amplitudes of the surface densities of current and charge. Despite this, it is still difficult to use such a mathematical model for a CAD system inasmuch as it is necessary to determine six values of the quantities being sought at each point on the surface of the conducting components.

The calculation of the electromagnetic field of parallel current conductors having a cylindrical form [7] and axiosymmetric conductors and screens [8] is an exception.

The calculation of a field using the secondary source method is implemented in two stages. First, all the secondary sources whose effect is equivalent to that of a heterogeneous medium are found. Next, the field is calculated using existing formulas (1).

The secondary sources are localized in space, but the field itself is not bounded. Thus, when the secondary source method is used to obtain a numerical solution of a problem, the number of unknowns is reduced sharply, and the precision increases. The secondary sources are natural determinant unknowns through which all remaining unknowns may be easily determined by using the formulas.

One big advantage of mathematical models derived by using the secondary source method is their invariance relative to the forms of the components of the design under consideration. This means that one and the same mathematical model may be used to design an entire class of devices with different geometries. For a numerical solution, the integral equations are replaced by an equivalent system of high-order algebraic equations. The number of computations needed to solve the algebraic system increases sharply as its order increases. An attempt should therefore be made to create a mathematical model in which the system of integral equations has the smallest possible dimensionality. To do this, only scalar quantities--electrical and magnetic charges--should be selected as secondary sources. This can always be done provided the medium is nonconductive.

The integral equations of the mathematical model are distinguished only by the dimensionality of their integrals, the structure of their nuclei, and the expressions in their right-hand part. For this reason, algorithms for making a numerical analysis of a mathematical model have a standard structure that makes it possible to automate the compilation of programs for calculating the complex electromagnetic fields of devices. The algorithm consists of blocks for computing the coefficients of a matrix, computing the vector of the right-hand part, forming the equation system and solving it, computing the field's potentials and vectors, and computing those integral parameters, the device's operating characteristics.

The secondary source method has served as the basis for creating a wide class of mathematical models in electrodynamics. It is used to develop a computer simulation of the fields in nonconductive media--linear part-homogeneous, part-nonlinear isotropic, part-nonlinear anisotropic. The field in each of these media can in turn by three dimensional, plane-parallel, and axiosymmetric.

As the capabilities of computer technology increase, the secondary source method will evidently find use in computer simulation of the electromagnetic fields in conductive linear and nonlinear media. The method is currently being used to calculate the electromagnetic field in conductive media, but virtually solely for two-dimensional fields.

The secondary source method has opened the possibility of improving mathematical models by introducing analytically discovered information about the properties of the distributions being solved for into the nucleus of the integral equation.

Figure 1 shows a part of the consolidated structure of a branch of an electrodynamics scientific knowledge base that is composed of numerical and analytical fragments and based on the secondary source method.

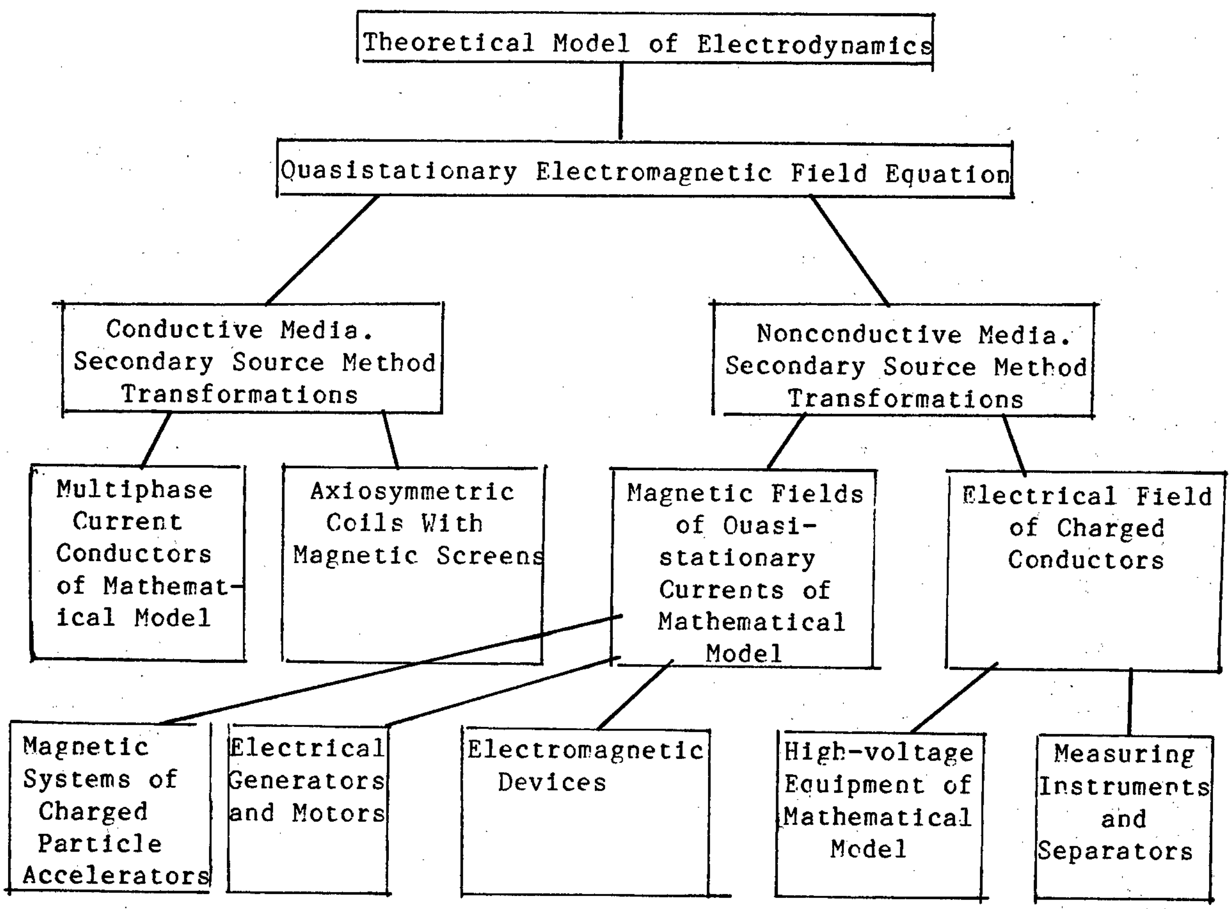


Figure 1.

Mathematical models of the electrodynamic theory of circuits. The next method of improving mathematical models is based on engineering experience and intuition. By analyzing the shapes of a device's components and the properties of their materials, it is sometimes possible to observe the influence of these shapes and properties on the structural features of the field in question and to introduce this information into the mathematical model. This information may not always be precise. When used correctly, however, the information makes it possible to simplify the mathematical model somewhat, which in turn results in a significant increase in the precision of numerical computation. The electrodynamic theory of circuits is an example of such a method that is used in conjunction with the secondary source method.

Electromagnetic systems consisting of a large number of componets, i.e., devices (multiport devices) in which electromagnetic processes that are accompanied by energy conversions occur and conductors (communicators) that connect multiport devices and direct the movement of energy, are widely used in modern equipment. The classic theories of circuits and uniform lines is generally used in designing such systems. However, these theories frequently do not provide the required precision. If, for example, a system's conductors are curved, branched, or crossed, if the medium surrounding them is heterogeneous, conductive, or ferromagnetic, or if the conductors in the different sections of the system neighboring systems are close to one another, then calculations based on the classic theories will be imprecise and unreliable. They overlook the electromagnetic inductions of the voltage and currents in curved and crossed conductors, the current leaks from conductors, energy losses in the medium, and radiation. At the same time, the trend in the development of equipment is toward increasing the frequency of currents in systems and assembling their components more compactly, which has resulted in an intensification of the electromagnetic interactions between all the sections of the systems--something that the classic theories do not take into account.

In contrast electrodynamic theory of circuits [9-11], which is based solely on currents and potentials, describes the electromagnetic process occurring in the system by taking into account the electromagnetic interactions between its components and the influence of the surrounding medium. The specific features this class of system designs, specifically, the length of the conductors (which results in an almost linear distribution of the field's sou ces) and the electromagnetic autonomy of the components, constitute the objective condition for the formulation of such a theory.

In the theory of electrical circuits the variety of components in a system are represented as one generalized circuit component--a multiport device. Similarly, the entire range of real designs of connecting conductors is represented in the electrodynamic theory of circuits by a generalized component--the communicator (i.e., a set of conductors and the surrounding dielectrics and screens that have been included between the multiport devices).

Unlike the multiport devices, communicators frequently do not have electromagnetic autonomy. The electromagnetic process occurring in each of them cannot always be separated from the electromagnetic process of the system as a whole. Unlike mulitport devices, which are standard components, when a systems is being designed it is necessary to work out an individual design for each communicator. It may be said that the process of designing an electromagnetic system comes down to selecting nominal values for its multiport devices and working out the designs for its communicators.

Theories of communicators in the form of uniform lines do exist [12, 13]; however, these are used only for the simplest designs and are not suitable for many of the forms of connections that are used in equipment. It has therefore become necessary to create a mathematical model of a communicator that is invariant relative to the shape of the connecting conductors and the structure of the medium surrounding them.

A mathematical model of a communicator has been derived by using analytical trnasformations of the theoretical model of electrodynamics. The following general properties of a field have been included in the model for the sake of simplification.

1. The system's external field, which determines the electromagnetic interactions of its componetns and energy dispersion in the surrounding medium, is created by the communicators' currents and charges.

2. The system's components (multiport devices) have electromagnetic autonomy, which means that the eigenfield is, for all practical purposes, localized within it, and its "contribution" to the system's external field is the same as would occur if a section of conductor with a length equal to that of the component had been connected instead of the component.

3. The elongated form of the conductors (small cross sections compared with their lengths) makes the distribution of currents and charges in them nearly linear. In other words, the field of a real communicator is little different from that of an ideal communicator with the same currents and linear charges but with infinitely thin conductors. In addition, the elongated form causes the current field in the conductor to have a unidimensional structure and results in a simple connection between the current and charge in the conductor. It is therefore possible to determine the electromagnetic process occurring inside the communicator if the distribution in only the current or linear charge is known.

4. The distinct difference between the electrical characteristics of the metal of the conductors and the medium results in a break in the current density vector. On the conductor's surface, the direction of the displacement current lines is perpendicular to that of the current lines in the conductor. Because the conductor is cylindrical in form, the displacement current density vectors on its surface lie in the plane of its cross section. Because of the smoothness of the field's potential functions, this planar structure of the field is maintained in a thin tubular layer of the medium surrounding the conductor [10]. Beyond the bounds of this tubular layer, the electromagnetic field generally has an essentially three-dimensional structure that becomes increasingly more complicated as the shape of the conductors' axes grows more complex.

5. Each small section of the currents elementary tube in the conductor is equivalent to the electrical dipole creating an electromagnetic field in the surrounding medium. Inasmuch as the medium is linear, the communicator's field may be found by superimposing the fields of all the dipoles filling the space of its conductors. This breaking the current in the circuit down into elementary dipoles makes it possible to calculate the effect of the heterogeneous medium surrounding the circuit with almost complete precision. In a heterogeneous medium with simple interfaces, a dipole's field may often be found analytically and its expression inserted into the equation for the communicator.

A mathematical model of the communicator, i.e., a complete system of functional equations relative to the distribution of current along the axes of its conductors, has been derived from the equations of the theoretical model by using the aforementioned features of the field's structure as supplementary information about the electromagnetic process occurring inside the communicator [9-11].

If in an elongated electromagnetic system the electromotive force and currents change over time in a sinusoidal manner, all the electrical quantities may be replaced by their complex transforms. The possibilities of the mathematical model are expanded in this particular case, which happens to be important from the standpoint of applications. The model becomes applicable to communicators with conductors located in partly homogeneous and weakly conductive media.

In the classic theory of circuits the presentation of multiport devices as concentrated components is based on a crude assumption of the independence of the electromagnetic process from the shape and dimensions of its components. In reality, the representation of a component (the source of electromotive force e or load z) as being concentrated is equivalent to its representation as being included in an infinitely thin cross section of the conductor. This type of idealization contributes to a disruption of the first kind in the distribution of the scalar potential and a sharp increase in the intensity of the electrical field and current in the vicinity of the concentrated component [12]. The representation of the branching of a conductor as being infinitely thin entails similar consequences. Consequently the real dimensions of all the components and branchings must be taken into account in the mathematical model.

Doing this requires replacing the real components of the electromagnetic system in the mathematical model by idealized local components. The term "local components" refers to an elongated component that is characterized by integral parameters (e, z) with electromagnetic autonomy and its own internal electrical field that is evenly distributed along the length Δ of the component. The length of the local component is assumed to be equal to the distance between the terminals of the real component. The sections Δ, which are replaced along the loop of the circuit by local components (e, z), and the coordinates of the centers of these sections are designated as $\Delta_э$, $\ell_э$, and Δ_N, l_N, respectively, and it is assumed that the internal electrical intensity is equally distributed along these sections and tangential to them and that the contribution of the local component to the circuit's external field is equal to that of a corresponding section with the very same current. At the place where the conductors branch in the section Δ_{Di} of a loop, which is equal to the linear dimension of the cross section of a conductor that is branched from the node, it is assumed that the current density is normal for this section and equally distributed along it.

The local components and branching are introduced into the mathematical model by using Π functions [11]. If n loads z_i and m sources of electromotive force e_i are included in the loop of an electromagnetic system and if there are q branches, then the integrodifferential equations of the electromagnetic system have the following form:

$$-\frac{d\varphi}{dl}=\left[R_0+\sum_{i=1}^{n} Z_i\Pi(l\Delta_{Ni})\right] I(l)-\sum_{i=1}^{m} e_i\Pi(l\Delta_{\text{э}i})+ \\ +j\omega\int_L I(l_P)\,\Psi_A(ll_P)\,dl_P+j\omega A_k(l); \tag{6}$$

$$-\frac{dI}{dl}=G\left[\varphi(l)-\int_L I(l_P)\,\Psi_\varphi(ll_P)\,dl_P\right]-G\varphi_k(l)+ \\ +\sum_{i=1}^{q} I(l_{Di})\,\Pi(l\Delta_{Di}), \tag{7}$$

where $G = (\nu + j\omega\varepsilon)g$, g is the linear geometric conductivity of the tubular layer of the medium surrounding the conductor; R_0 is the linear resistance of the conductor; L is the loop formed by the axes of all of the communicator's conductors; ν and ε are the electrical conduction and dielectric permeability of the medium; A_k and φ_k are the average values of the components of the delayed potentials caused by sources located outside the conductors of the communicator under consideration; and Ψ_A and Ψ_φ are functions of the source expressed through the vector and scalar delayed potentials of a single dipole in the partly-homogeneous medium. Their expressions for simple forms of media interfaces that are encountered in actual practice are derived analytically and are presented in work [2].

The equation system (6) and (7) is sufficiently precise to describe the current distribution in the linear circuit all the way up to very high frequencies. It has been proved that there is a unique solution to this system. It is not difficult to calculate the entire electromagnetic process after the system has been solved and the distribution of currents in the conductors and components of the circuit has been found.

It is not possible to obtain an analytical solution of equation system (6) and (7). Obtaining a result requires constructing a mathematical model and using numerical methods. To do this it is advisable to use the integro-differential equations (6) and (7) and transform them into integral equations of the second kind. The algorithm for such equations is more reliable from the standpoint of precision and requires less computer time.

The transformation of equations (6) and (7) is reduced to the sequential integration of the equations along the loop L of the circuit, which consists of the axes of all its branches, and the exclusion of the value of the potential. The potentials φ and φ_0 should be coordinated, and the boundary conditions at the terminals of all of the circuit's conductors should be satisfied during the process of transformation.

For an elongated electromagnetic system (a strip microwave device) with a branched communicator located on a dielectric base with a screen containing a source of electromotive force, four-terminal networks, and loads connected to a grounded metal screen, the order of actions when transforming system (6) and (7) to integral equations is as follows. First, integrate equation (6) beginning from the system's entry point (i.e., the point where the electromotive force is connected). Then, one after the other, find expressions for the potentials of the moving points of the initial branch, the intermediate and end branches, and the potentials of the connection points of the conductions of the local four-terminal networks and load resistances. The voltage drops in the resistances of the local four-terminal networks located on the path of the integration are taken into account in this process. Then, the type of integral equations for each branch is determined by integrating equation (7) in the reverse sequence--beginning from the system's exit points. Expressions are used for currents that branch through the conductions of the local four-terminal networks located on the path of the integration to the screen, whose potential φ is assumed to be equal to zero.

Techniques have currently been developed to transform equations encompassing all communicator designs and sequences of connecting multiport devices that are encountered in practice [14]. As a result of the transformation, the mathematical model acquires the form of a system of integral equations of the second type, each of which can be converted to the following form:

$$I(l_Q) + \int_L I(l_P)\, K(l_Q l_P)\, dl_P = f(l_Q, e, I_k), \tag{8}$$

where e and I_k are the specified electromotive force and current of the system's power supplied; and $K(l_Q,l_p)$ is the equation's nucleus, which in the general case is expressed through curvilinear integrals of the functions Ψ_A and Ψ_φ and the components' parameters.

The electrodynamic theory of circuits served as the basis for creating a wide class of mathematical models that may be used to develop a computer simulation of the electromagnetic processes occurring in the following: a network of electric railways and underground pipelines (to protect the latter against corrosion caused by stray currents); directional emission antennas and phased antenna arrays (to determine their directional patterns and emission powers); strip devices and microwave systems (to design them reliably); integrated circuits used in computer equipment (to determine electromagnetic compatibility and functional efficiency), etc.

Figure 2 shows a portion of the consolidated structure of a branch of an electrodynamics knowledge base that consists of numerical and analytical fragments based on the electrodynamic theory of circuits.

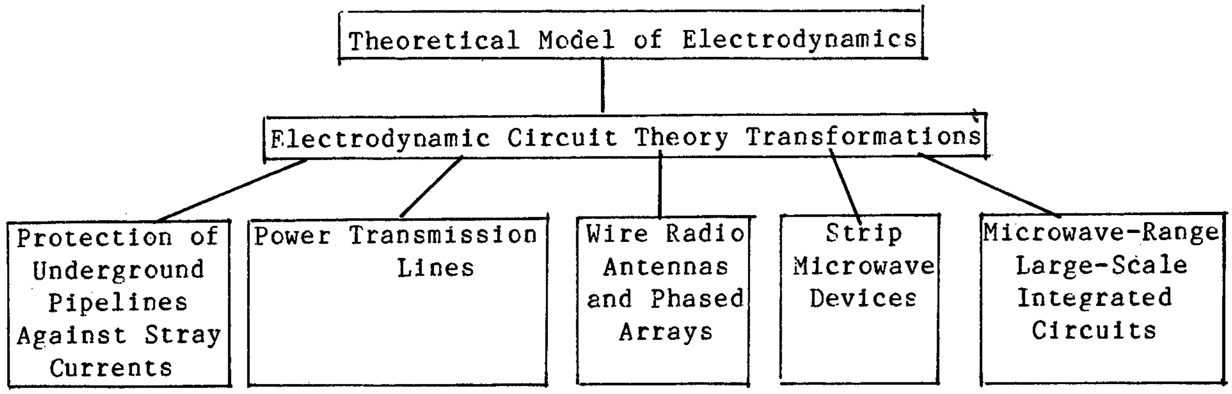


Figure 2.

The mathematical model for each of the aforementioned types of devices has its own specific features, which are determined by the design features of its communicators and its function. For example, the stray currents of railways and electrical power installations which are constant in time, permeate into underground pipelines through insulation and cause the pipelines' metal to corrode. The intensity of the destruction of the metal in the pipes is determined by the magntidue of the leakage current proportional to the voltage on the insulation. If the distribution of voltage along the pipelines is found, it is possible to position protective devices to eliminate the corrosion.

To do this, equations (6) and (7) are transformed so that the variable being solved for is the value of the voltage $u(1) = \varphi(1) - \varphi_0(1)$ on the pipes' insulation. Assuming that ω, Z, and e are all equal to zero and adding the quantity $\frac{d\varphi_0}{dl}$ to both sides of (6), we obtain the following equation system:

$$-\frac{du}{dl} = R_0 I(l) + \frac{d\varphi_0}{dl}, \qquad (9)$$
$$-\frac{dI}{dl} = gu(l) + \sum_{i=1}^{q} I(l_{Di})\,\Pi(l\Delta_{Di}).$$

The sources of the potential φ_0 are the currents flowing from all the grounds of the electrical power installations and the leak currents from the railway's rails and pipelines. The potential component φ_k of the current field of all the grounds is known. The component created by the leak currents is determined by the voltage u(1), which is being sought. Consequently, the loop potential is specified as

$$\varphi_0(Q) = \int_L g(l)\,u(l)\,\Psi(l_Q l)\,dl + \varphi_k(l), \qquad (10)$$

where $\Psi(l_0 l)$ is the electrical potential of a point source of unit value located in the soil. If the boundary of the rock foundation of the soil is assumed to be flat, the value of Ψ may be determined by using the mirror image method [13].

The equation system (9) and (10) with boundary conditions at the pipe ends is sufficient to formulate any concrete problem correctly. To derive a mathematical model from the system, equations (9) and (10) must be transformed into integral equations.

A second example is the specific features of the mathematical model of a microwave device based on microstrip lines. The design of the device's communicator is distinguished by the layered structure of the surrounding medium (base and screen) and the elongated rectangular shape of a [cross] section of the conductor. The device's functioning is distinguished by the microwave frequency of its current flows.

The effect of the layered medium on the microwave-frequency electromagnetic process is taken into account by using source functions [2], i.e., functions of the delayed potentials of the electromagnetic field of the single radiating dipole. However, a simple substitution of the strip conductor by a filament in the expressions for the loop potentials may result in a significant error if the size of the strip's cross section is commensurate with the length of the electromagnetic field in air. To reduce the error, each source function is averaged from the standpoint of the width of the strip conductors and the external loop of the tubular layer of the planar field [15]. Despite the apparent cumbersome nature of the integral equations in (8), their numerical solution on a computer does not present a problem. They have a stable solution and contain only unidimensional integrals that can be calculated on a computer with a high degree of precision.

The proposed mathematical model of an elongated electromagnetic system is an approximate model. It uses the results of a qualitative analysis of the structure of the electromagnetic field, proceeds from approximate distributions of the charge and current in the cross section of the conductors, and is based on loop potentials rather than on real potentials. In this manner it dramatically simplifies the task of calculating an electromagnetic process by preserving the principal information about the geometry of the conductors and the structure of the surrounding medium, and the manner in which it takes all of the principal electromagnetic connections between conductors into account is correct. The more adequately the mathematical model reflects the real electromagnetic process and the more precisely it specifies the loop potentials and parameters of the tubular layer in its equations, the greater its precision. These conditions depend on the shape and relationships of the dimensions of the communicator's conductors. Specifically, the smaller the diameter of the conductors' cross sections compared with the distance between their axes, radius of curvature, and wavelength in the surrounding dielectric, the more precise the mathematical model.

The estimates presented here have shown that if the distance between the conductors' axes and the axes' radius of curvature exceeds twice the diameter of their cross sections and the wavelength in the dielectric is at least fivefold greater than the diameter, the model's precision is adequate for practical use.

BIBLIOGRAPHY

1. Tozoni, O. V., "Content and Function of Subject Area Knowledge Bases for Automated Scientific Research and Computer-Aided Design of New Equipment," KIBERNETIKA, No 1, 1986, pp 52-63, 66.

2. Tozoni, O. V., "Metod vtorichnykh istochnikov v elektrotekhnike" [Secondary Source Method in Electrical Engineering], Moscow, Energiya, 1975, 295 pages.

3. Mamedshakhov, M. E., Nezhinskaya, M. M., Nikolayeva, N. S., and Tozoni, O. V., "Constructing Computational Model of Field in Gap of Linear Asynchronous Motor," ELEKTRICHESTVO, No 10, 1985, pp 32-40.

4. Tozoni, O. V., "Analytical Calculation of Electromagnetic Process in Linear Motor," IZV. AN SSSR, SER. ENERGETIKA I TRANSPORT, No 5, 1977, pp 100-114.

5. Korn, G., and Korn, T., "Spravochnik po matematike dlya nauchnykh rabotnikov i inzhenerov" [Mathematics Manual for Scientific Workers and Engineers], Moscow, Nauka, 1968, 720 pages.

6. Tozoni, O. V., and Mayergoyz, I. D., "Raschet trekhmernykh elektromagnitnykh poley," [Calculation of Three-Dimensional Electromagnetic Fields], Kiev, Tekhnika, 1974, 310 pages.

7. Tozoni, O. V., "Raschet elektromagnitnykh poley na vychislitelnykh mashinakh," [Calculation of Electromagnetic Fields on Computers], Kiev, Tekhnika, 1967, 252 pages.

8. Bozhenko, A. I., Petrushenko, Ye. I., Royzman, P. S., and Ryskov, V. S., "Using Digital Computer to Model Eddy Currents in Axiosymmetric Non-magnetic Conductors With Allowance for Symmetry of Integral Equation," ELEKTRON. MODELIROVANIYE, No 4, 1982, pp 33-41.

9. Tozoni, O. V., "Principles of Electrodynamic Theory of Circuits," ELEKTRICHESTVO, No 7, 1980, pp 40-47.

10. Tozoni, O. V., "Electrodynamic Theory of Circuits," IZV. AN SSSR, SER. ENERGETIKA I TRANSPORT, No 1, 1977, pp 85-100

11. Tozoni, O. V., "Equations for Electrodynamic Theory of Circuits," Ibid., 1979, pp 67-85.

12. Vaynshteyn, L. A., "Elektromagnitnyye volny" [Elektromagnetic Waves], Moscow, Sov. radio, 1957, 450 pages.

13. Kostenko, M. V., Perelman, L. S., and Shkarik, Yu. P., "Volnovyye protsessy i elektricheskiye pomekhi v mnogoprovodnykh liniyakh vysokogo napryazheniya" [Wave Processes and Electrical Noise in Multiconductor High-Voltage Lines], Moscow, Energiya, 1973, 320 pages.

14. Tozoni, O. V., Indichenko, L. M., Vasilyev, V. Ye., and Glachneko, V. V., "Principles of Modeling Electromagnetic Processes in Extended Systems," ELEKTRON. MODELIROVANIYE, No 2, 1979, pp 53-67.

15. Tozoni, O. V., and Vasilyev, V. Ye., "Calculation of Current Distribution in Treelike Schemes of Microwave Circuits With Concentrated Four-Terminal Networks," KIBERNETIKA I VYCHISL. TEKHNIKA, No 30, 1975, pp 19-40.

12794/9835 - END -